



**HAL**  
open science

# Gestion des stocks dans des chaînes logistiques face aux aléas des délais d'approvisionnements

Faïcel Hnaien

► **To cite this version:**

Faïcel Hnaien. Gestion des stocks dans des chaînes logistiques face aux aléas des délais d'approvisionnements. Génie des procédés. Ecole Nationale Supérieure des Mines de Saint-Etienne, 2008. Français. NNT : 2008EMSE0037 . tel-00784217

**HAL Id: tel-00784217**

**<https://tel.archives-ouvertes.fr/tel-00784217>**

Submitted on 4 Feb 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre : 502 GI

**THESE**  
présentée par

*Faïcel Hnaien*

Pour obtenir le grade de Docteur  
de l'Ecole Nationale Supérieure des Mines de Saint-Étienne

Spécialité : Génie industriel

*Gestion des stocks dans des chaînes logistiques face aux aléas  
des délais d'approvisionnements*

Soutenance prévue à l'Ecole des Mines de Saint Etienne le 08/12/2008

**Membres du jury**

*Rapporteurs :*

Marie Claude Portmann  
Jean-Pierre Campagne

Professeur, Ecole des Mines de Nancy  
Professeur, INSA Lyon

*Examineurs :*

Bernard Grabot  
Jean-Claude Hennet  
Pierre Ladet

Professeur, ENI de Tarbes  
Directeur de recherche au CNRS, Université Paul Cézanne  
Professeur, ENS d'Ingénieurs Electriciens de Grenoble

*Directeur de thèse :* Alexandre Dolgui  
*Co-encadrant de thèse :* Hélène Marian  
*Co-encadrant de thèse :* Mohamed Aly Ould-Louly

Professeur, Ecole des Mines de Saint Etienne  
Maître assistant, Ecole des Mines Saint Etienne  
Associate professor, King Saud University, Arabie Saudi



**Spécialités doctorales :**

SCIENCES ET GENIE DES MATERIAUX  
 MECANIQUE ET INGENIERIE  
 GENIE DES PROCÉDES  
 SCIENCES DE LA TERRE  
 SCIENCES ET GENIE DE L'ENVIRONNEMENT  
 MATHÉMATIQUES APPLIQUÉES  
 INFORMATIQUE  
 IMAGE, VISION, SIGNAL  
 GENIE INDUSTRIEL  
 MICROELECTRONIQUE

**Responsables :**

J. DRIVER Directeur de recherche – Centre SMS  
 A. VAUTRIN Professeur – Centre SMS  
 G. THOMAS Professeur – Centre SPIN  
 B. GUY Maître de recherche – Centre SPIN  
 J. BOURGOIS Professeur – Centre SITE  
 E. TOUBOUL Ingénieur – Centre G2I  
 O. BOISSIER Professeur – Centre G2I  
 JC. PINOLI Professeur – Centre CIS  
 P. BURLAT Professeur – Centre G2I  
 Ph. COLLOT Professeur – Centre CMP

**Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)**

AVRIL	Stéphane	MA	Mécanique & Ingénierie	CIS
BATTON-HUBERT	Mireille	MA	Sciences & Génie de l'Environnement	SITE
BENABEN	Patrick	PR 2	Sciences & Génie des Matériaux	CMP
BERNACHE-ASSOLANT	Didier	PR 1	Génie des Procédés	CIS
BIGOT	Jean-Pierre	MR	Génie des Procédés	SPIN
BILAL	Essaïd	DR	Sciences de la Terre	SPIN
BOISSIER	Olivier	PR 2	Informatique	G2I
BOUCHER	Xavier	MA	Génie Industriel	G2I
BOUDAREL	Marie-Reine	MA	Sciences de l'inform. & com.	DF
BOURGOIS	Jacques	PR 1	Sciences & Génie de l'Environnement	SITE
BRODHAG	Christian	MR	Sciences & Génie de l'Environnement	SITE
BURLAT	Patrick	PR 2	Génie industriel	G2I
CARRARO	Laurent	PR 1	Mathématiques Appliquées	G2I
COLLOT	Philippe	PR 1	Microélectronique	CMP
COURNIL	Michel	PR 1	Génie des Procédés	SPIN
DAUZERE-PERES	Stéphane	PR 1	Génie industriel	CMP
DARRIEULAT	Michel	ICM	Sciences & Génie des Matériaux	SMS
DECHOMETS	Roland	PR 1	Sciences & Génie de l'Environnement	SITE
DESRAYAUD	Christophe	MA	Mécanique & Ingénierie	SMS
DELAFOSSÉ	David	PR 1	Sciences & Génie des Matériaux	SMS
DOLGUI	Alexandre	PR 1	Génie Industriel	G2I
DRAPIER	Sylvain	PR 2	Mécanique & Ingénierie	SMS
DRIVER	Julian	DR	Sciences & Génie des Matériaux	SMS
FOREST	Bernard	PR 1	Sciences & Génie des Matériaux	CIS
FORMISYN	Pascal	PR 1	Sciences & Génie de l'Environnement	SITE
FORTUNIER	Roland	PR 1	Sciences & Génie des Matériaux	CMP
FRACZKIEWICZ	Anna	MR	Sciences & Génie des Matériaux	SMS
GARCIA	Daniel	CR	Génie des Procédés	SPIN
GIRARDOT	Jean-Jacques	MR	Informatique	G2I
GOEURIOT	Dominique	MR	Sciences & Génie des Matériaux	SMS
GOEURIOT	Patrice	MR	Sciences & Génie des Matériaux	SMS
GRAILLOT	Didier	DR	Sciences & Génie de l'Environnement	SITE
GROSSEAU	Philippe	MR	Génie des Procédés	SPIN
GRUY	Frédéric	MR	Génie des Procédés	SPIN
GUILHOT	Bernard	DR	Génie des Procédés	CIS
GUY	Bernard	MR	Sciences de la Terre	SPIN
GUYONNET	René	DR	Génie des Procédés	SPIN
HERRI	Jean-Michel	PR 2	Génie des Procédés	SPIN
KLÖCKER	Helmut	MR	Sciences & Génie des Matériaux	SMS
LAFOREST	Valérie	CR	Sciences & Génie de l'Environnement	SITE
LI	Jean-Michel	EC (CCI MP)	Microélectronique	CMP
LONDICHE	Henry	MR	Sciences & Génie de l'Environnement	SITE
MOLIMARD	Jérôme	MA	Sciences & Génie des Matériaux	SMS
MONTHEILLET	Frank	DR 1 CNRS	Sciences & Génie des Matériaux	SMS
PERIER-CAMBY	Laurent	PR 1	Génie des Procédés	SPIN
PIJOLAT	Christophe	PR 1	Génie des Procédés	SPIN
PIJOLAT	Michèle	PR 1	Génie des Procédés	SPIN
PINOLI	Jean-Charles	PR 1	Image, Vision, Signal	CIS
STOLARZ	Jacques	CR	Sciences & Génie des Matériaux	SMS
SZAFNICKI	Konrad	CR	Sciences de la Terre	SITE
THOMAS	Gérard	PR 1	Génie des Procédés	SPIN
VALDIVIESO	François	MA	Sciences & Génie des Matériaux	SMS
VAUTRIN	Alain	PR 1	Mécanique & Ingénierie	SMS
VIRICELLE	Jean-Paul	MR	Génie des procédés	SPIN
WOLSKI	Krzysztof	CR	Sciences & Génie des Matériaux	SMS
XIE	Xiaolan	PR 1	Génie industriel	CIS

**Glossaire :**

PR 1	Professeur 1 <sup>ère</sup> catégorie
PR 2	Professeur 2 <sup>ème</sup> catégorie
MA(MDC)	Maître assistant
DR (DR1)	Directeur de recherche
Ing.	Ingénieur
MR(DR2)	Maître de recherche
CR	Chargé de recherche
EC	Enseignant-chercheur
ICM	Ingénieur en chef des mines

**Centres :**

SMS	Sciences des Matériaux et des Structures
SPIN	Sciences des Processus Industriels et Naturels
SITE	Sciences Information et Technologies pour l'Environnement
G2I	Génie Industriel et Informatique
CMP	Centre de Microélectronique de Provence
CIS	Centre Ingénierie et Santé

# Table des matières

<b>Introduction générale</b> .....	<b>1</b>
<b>Chapitre 1 : Généralités sur la gestion des stocks et la planification des chaînes logistiques</b> .....	<b>4</b>
1.1 Introduction.....	4
1.2 Chaînes logistiques « Supply chain » .....	5
1.2.1 Définition .....	5
1.2.2 Structures possibles de la chaîne logistique.....	6
1.3 La gestion des stocks .....	7
1.3.1 Les paramètres de la gestion des stocks.....	8
1.3.2 Origine et nature des stocks dans une chaîne logistique.....	9
1.3.3 Les politiques classiques de gestion des stocks .....	9
1.4 La politique de planification de type MRP .....	14
1.4.1 Principe de la méthode MRP .....	15
1.4.2 Les incertitudes sous MRP .....	17
1.4.3 Stock et délai de sécurité.....	19
1.4.4 Méthodes de lotissement (lot-sizing).....	21
1.5 Problématique de la thèse .....	23
1.6 Conclusion .....	24
<b>Chapitre 2 : Etat de l'art</b> .....	<b>25</b>
2.1 Introduction.....	25
2.2 Chaîne d'approvisionnement à structure linéaire.....	27
2.2.2 Structure à plusieurs niveaux.....	29
2.2.3 Perspectives de recherche .....	31
2.3 Systèmes d'assemblage.....	33
2.3.1 Structure à un niveau .....	33
2.3.2 Structure multi-niveau .....	37
2.3.3 Perspectives de recherche .....	39
2.4 Choix de fournisseurs.....	41
2.5 Chevauchement des ordres (« crossover »).....	42
2.6 Conclusions.....	44
<b>Chapitre 3 : Chaîne d'approvisionnements linéaire</b> .....	<b>47</b>
3.1 Introduction.....	47
3.2 Le cas de fabrication au plus tôt.....	48
3.2.1 Problème P1.....	48
3.2.1.1 Description du problème .....	49
3.2.1.2 Minimisation des coûts .....	50
3.2.1.3 Optimisation.....	52
3.2.1.4 Exemples numériques .....	53
3.2.2 Problème P2.....	55
3.2.2.1 Description du problème .....	55
3.2.2.2 Minimisation des coûts .....	56
3.2.2.3 Optimisation.....	57
3.2.2.4 Exemples numériques .....	57
3.3 Le cas avec dates de fabrication.....	58
3.3.1 Problème P3.....	59
3.3.1.1 Description du problème P3.....	59
3.3.1.2 Minimisation des coûts .....	60
3.3.1.3 Optimisation.....	64
3.3.1.4 Exemples numériques .....	72
3.3.2 Problème P4.....	74

3.3.2.1 Description du problème .....	74
3.3.2.2 Minimisation du coût moyen .....	75
3.3.2.3 Optimisation.....	76
3.3.2.4 Tests numériques.....	76
3.4 Conclusion .....	76
<b>Chapitre 4 : Optimisation des systèmes d'assemblage à deux niveaux par une PSE .....</b>	<b>78</b>
4.1 Introduction.....	78
4.2 Description du problème .....	79
4.3 Problème PP1 .....	81
4.3.1 Description du problème .....	81
4.3.2 Minimisation des coûts .....	82
4.3.3 Optimisation .....	85
4.3.3.1 Problèmes à résoudre .....	85
4.3.3.2 Pré-optimisation : procédure de réduction de l'espace de recherche.....	86
4.3.3.3 Accroissements partiels .....	93
4.3.3.4 Propriétés de dominance .....	96
4.3.3.5 Bornes inférieures .....	98
4.3.3.6 Borne supérieure .....	100
4.3.3.7 Algorithme PSE .....	101
4.3.4 Exemples Numériques .....	102
4.3.4.1 Impact de la pré-procédure de réduction de l'espace de recherche des solutions.....	102
4.3.4.2 Comportement de PSE.....	103
4.3.4.3 Tests des performances de l'algorithme.....	105
4.4 Problème PP2.....	106
4.4.1 Description du problème .....	106
4.4.2 Minimisation des coûts .....	107
4.4.3 Optimisation .....	109
4.4.3.1 Problèmes à résoudre .....	109
4.4.3.2 Pré-procédure de réduction de l'espace de recherche de solution .....	109
4.4.3.3 Propriétés de dominance .....	111
4.4.3.4 Bornes inférieures .....	113
4.4.3.5 Borne supérieure .....	114
4.4.3.6 Algorithme PSE .....	114
4.4.4 Exemples Numériques .....	115
4.4.4.1 Impact de la pré-procédure de réduction de l'espace de recherche des solutions.....	115
4.4.4.2 Etude numérique de l'algorithme.....	116
4.4.4.3 Comportement de PSE.....	116
4.4.4.4 Tests des performances des algorithmes.....	117
4.5 Conclusions.....	118
<b>Chapitre 5 : Optimisation d'un système d'assemblage à deux niveaux par un algorithme génétique .....</b>	<b>119</b>
5.1 Introduction.....	119
5.2 Algorithme génétique.....	120
5.2.1 Codage des solutions et fonction Fitness .....	124
5.2.1.1 Codage des solutions.....	124
Chaque chromosome de l'algorithme génétique est évalué à travers une fonction appelée fonction Fitness. Dans la section suivante nous définissons cette fonction plus en détail. ....	124
5.2.1.2 Fonction Fitness .....	124
5.2.2 Génération de la population initiale .....	125
5.2.3 Sélection.....	126
5.2.4 Opérateur de croisement .....	127
5.2.5 Opérateur de mutation .....	128
5.2.6 Recherche locale .....	129

5.3 Résultats expérimentaux .....	130
5.3.1 Conditions expérimentales .....	131
5.3.2 Résultats expérimentaux de l'algorithme génétique .....	132
5.4.2 Impact de la recherche locale .....	135
5.5 Conclusion .....	137
<b>Conclusions générales .....</b>	<b>139</b>

# Liste des figures

Figure 1.1 : Chaîne logistique .....	5
Figure 1.2 : Structures possibles d'une chaîne logistique.....	6
Figure 1.3 : Exemples des chaînes logistiques à nomenclature multi-niveau .....	7
Figure 1.4 : Politique d'approvisionnement à point de commande .....	10
Figure 1.5 : Politique d'approvisionnement à quantité économique .....	11
Figure 1.6 : Illustration de la relation entre les délais d'approvisionnement planifié et réel.....	13
Figure 1.7 : Nomenclature de produit fini .....	17
Figure 1.8 : Plan Directeur de Production (PDP) .....	17
Figure 1.9 : Incertitudes des données pour la méthode MRP .....	18
Figure 1.10 : Paramètres de du MRP.....	19
Figure 3.1 : Chaîne d'approvisionnement linéaire à $m$ niveaux .....	47
Figure 3.2 : Cas de fabrication au plus tôt.....	48
Figure 3.3 : Description du problème P1.....	49
Figure 3.4 : Chaîne d'approvisionnement linéaire à 10 niveaux .....	54
Figure 3.5 Description du problème P2.....	56
Figure 3.6 : Cas avec dates de fabrication .....	58
Figure 3.7 : Description du problème P3.....	60
Figure 3.8 : Description du problème P4.....	74
Figure 4.1 : Exemple d'un système d'assemblage à deux niveaux .....	78
Figure 4.2 : Description du problème PP1 .....	81
Figure 4.3 : Décomposition du problème PP1.....	87
Figure 4.4 : L'évolution de la borne inférieure pour le problème PP1 .....	104
Figure 4.5 : L'évolution de nombre des sous-ensembles d'arbre de recherche pour le problème PP1 .....	105
Figure 4.6 : Description du problème PP2 .....	107
Figure 4.7 : l'évolution de nombre des noeuds de l'arbre de recherche pour le problème PP2 .....	117
Figure 5.1 : Les principes fondamentaux d'un algorithme génétique .....	122
Figure 5.2 : Algorithme génétique pour le problème PP1 .....	123
Figure 5.3 : Représentation de chromosome .....	124
Figure 5.4 : Opérateur de croisement .....	128
Figure 5.5 : Procédure pour obtenir le meilleur voisinage de la solution.....	130
Figure 5.6 : La procedure de Hill-Climbing .....	131
Figure 5.7 : L'écart entre la meilleure et le pire run de chaque instance.....	133
Figure 5.8 : La performance moyenne en fonction de la taille des problèmes .....	134
Figure 5.9 : Le temps de calcul "CPU " en fonction de la taille de problème.....	135
Figure 5.10 : Le temps CPU moyen des algorithmes AG et AG+RL .....	136
Figure 5.11 : Exemple de convergence de la population pour un run d'une instance de taille 120 .....	137

# Liste des tableaux

Tableau 2.1 : Classification des résultats pour le cas de chaînes à structure linéaire.....	32
Tableau 2.2 : Classification des résultats pour le cas des systèmes d'assemblage.....	40
Tableau 3.1 : La date optimale de lancement d'ordre de fabrication en fonction de coût de retard.....	54
Tableau 3.2 : La date de lancement d'ordre optimale en fonction de nombre de niveaux.....	55
Tableau 3.3 : Coût de stockage, de replanification et de rupture.....	72
Tableau 3.4 : Loi de probabilité des délais d'approvisionnement.....	72
Tableau 3.5 : Les limites inférieures et supérieures.....	73
Tableau 3.6 : La solution optimale.....	73
Table 4.1 : La probabilité de distribution des délais d'approvisionnement.....	102
Table 4.2 : Les coûts de stockages unitaires.....	103
Tableau 4.3 : Les limites inférieures et supérieures.....	103
Table 4.4 : La probabilité de distribution des délais d'approvisionnement.....	104
Table 4.5 : Le temps moyen de calcul (en seconds).....	106
Table 4.6 : Les limites inférieures et supérieures des variables de décision.....	115
Table 4.7 : Le temps moyen de calcul (en seconds).....	118
Table 5.1 : Performances moyennes des algorithmes génétiques (AG) et (AG + RL).....	135
Table 5.2 : Le temps moyen de convergence des AG et AG+RL (en secondes).....	136

# Introduction générale

La gestion de production a pour objet la recherche d'une organisation efficace de la production de biens et de service. En situant la production dans la perspective plus large de la chaîne logistique (*supply chain*), la définition de cette organisation doit aussi impérativement prendre en compte la maîtrise des flux entrants et sortants, autrement dit de l'approvisionnement jusqu'à la distribution pour assurer le niveau voulu de satisfaction de clients.

Face à une intensification de la concurrence, la gestion des systèmes de production a beaucoup changé ces dernières années. Les industriels sont toujours sollicités à innover en optimisant l'organisation de leurs entreprises, à prendre les bonnes décisions au bon moment afin de satisfaire leurs clients à moindre coût. Les chercheurs sont à leur tour essayent de fournir aux industriels des solutions nouvelles et efficaces.

Pour mieux situer les différents problèmes rencontrés en gestion de la production, les décisions de gestion sont classées en trois catégories : décisions stratégiques, tactiques et opérationnelles. Les décisions stratégiques se traduisent par la formulation d'une politique à long terme de l'entreprise qui concerne le choix des fournisseurs, des ressources, d'un mode transport, etc. Les décisions tactiques correspondent à un ensemble des décisions à moyen terme. Parmi les décisions tactiques on trouve la planification de la production. Les décisions opérationnelles assurent la flexibilité quotidienne nécessaire pour faire face aux fluctuations prévues de la demande et des délais et permettent de réagir face aux aléas dans le respect des décisions tactiques. Parmi les décisions opérationnelles on trouve la gestion des stocks et l'ordonnancement. Ce travail de thèse s'intègre dans la problématique de gestion des stocks dans une chaîne logistique face aux aléas des délais d'approvisionnements. Ce travail se situe donc au niveau de décisions opérationnelles.

Différentes sources d'aléas existent le long de la chaîne logistique telle que la demande en produit fini et les délais d'approvisionnement en composants. La gestion des stocks en présence de ces aléas est un problème classique pour les entreprises industrielles. Le plus souvent, cette gestion des stocks est élaborée en considérant des variations possibles sur la demande. En effet, beaucoup de travaux existent sur la planification des réapprovisionnements pour ce type d'aléa. Les modèles qui considèrent des variations sur les autres données du problème sont plus rares, notamment les modèles qui prennent en compte la variabilité des délais d'approvisionnement.

Pourtant, ces délais sont rarement constants, différents événements plus au moins prévisibles le long de la chaîne logistique peuvent causer des perturbations (panne de machine, problème de transport, qualité,...).

En effet, une mauvaise politique de gestion des approvisionnements conduit soit à des retards de livraison, qui engendrent des frais, soit à des stocks inutiles. Ces derniers peuvent être créés à différents niveaux (des matières premières aux produits finis), coûtent de l'argent et immobilisent des fonds. C'est pourquoi, il faut être vigilant et adopter des méthodes de gestion des stocks et de planification des approvisionnements efficaces pour savoir quoi commander, combien et quand.

Une difficulté complémentaire de la planification des réapprovisionnements des chaînes logistiques est liée à la structure de la chaîne (en plus des problèmes liés aux aléas). En effet, dans le cas où il y a de l'assemblage, plusieurs composants sont nécessaires pour fabriquer un produit (fini ou semi-fini) ce qui crée une interdépendance entre les stocks de ces composants. Egalement pour les chaînes logistiques dont la nomenclature du produit fini est à plusieurs niveaux, il y a une interdépendance entre les stocks de différents niveaux, car une perturbation (retard ou avance) à un niveau donné crée une perturbation au niveau suivant.

Le sujet de la thèse a été donc choisi comme l'étude des problèmes de gestion des stocks et de la planification des réapprovisionnements pour différentes structures de chaînes logistiques à plusieurs niveaux, soumises aux aléas des délais d'approvisionnements. La thèse se trouve donc au cœur des sujets actuels de la gestion de la production.

Etant donné la complexité du problème, nous nous sommes limités à la planification des réapprovisionnements pour des chaînes logistiques avec un seul type de produit fini et sur une seule période, c'est-à-dire pour une date et volume connus. Pour ce cas, nous avons démontré des propriétés théoriques intéressantes et nous avons proposé des méthodes efficaces de planification.

Nous avons choisi comme variables de décision celles qui correspondent aux temps de cycle planifiés qui sont des paramètres de la méthode MRP, l'objectif pratique de notre étude étant également de fournir des réflexions et des méthodes de paramétrage des logiciels MRP en présence des aléas des délais d'approvisionnements.

L'exposé de nos travaux est organisé en cinq chapitres.

Le premier chapitre présente la problématique de gestion des stocks et de planification de réapprovisionnements. Nous présentons une chaîne logistique d'une façon générale. Un intérêt particulier est porté aux différentes structures d'une chaîne logistique et aux problèmes de réapprovisionnements correspondants. Ensuite, ces problèmes sont déclinés pour la planification de type MRP. La démarche globale de cette méthode est présentée ainsi que les incertitudes sous MRP.

Le chapitre 2 est un état de l'art sur les principaux travaux qui s'intéressent aux incertitudes des délais d'approvisionnements. Nous présentons une classification des problèmes en nous basant sur la structure de la chaîne logistique : les systèmes à un seul fournisseur, les systèmes à structure linéaire et les systèmes d'assemblage à un et à plusieurs niveaux. Nous avons également donné une classification des méthodes proposées : les méthodes approchées et les méthodes exactes ainsi que des modèles utilisés à variables continues et à variables discrètes. Enfin, nous avons présenté des diverses problématiques et perspectives.

Le chapitre 3 s'intéresse à la planification des réapprovisionnements pour le cas des chaînes à structure linéaire multi-niveau. Nous avons étudié les deux cas suivants : le premier cas concerne la fabrication juste à temps où il n'y a que le coût de stockage des composants de niveau 1 et le coût de rupture pour le produit fini ; et le second cas où il y a un coût de stockage et un coût de replanification à chaque niveau et un coût de rupture en produit fini. Nous avons choisi deux critères d'optimisation. Le premier minimise le coût moyen total composé du coût moyen de stockage des composants et du coût moyen de rupture en produit fini. Le deuxième est la minimisation de coût moyen de stockage sous la contrainte de niveau de service. Nous offrons une présentation analytique de la fonction coût à minimiser pour les deux cas et une méthode récursive d'optimisation basée sur le modèle connu de marchand des journaux (*Newsboy model*).

Dans le chapitre 4, nous nous intéressons à la planification des réapprovisionnements des systèmes d'assemblage. Nous offrons une modélisation analytique du problème pour le cas des systèmes d'assemblage à deux niveaux. L'objectif est de minimiser le coût total moyen composé : i) du coût moyen de stockage des composants et du coût moyen de rupture en produit fini et ii) du coût moyen de stockage des composants sous la contrainte de niveau de service du client final. Nous présentons une méthode exacte d'optimisation basée sur une procédure par séparation et évaluation. Les résultats expérimentaux montrent que cette méthode est limitée à des problèmes de taille moyenne.

Enfin, nous proposons au chapitre 5 un algorithme génétique qui permet de résoudre le même problème que celui de chapitre 4 (système d'assemblage à deux niveaux) mais pour les instances de taille plus importante. D'autres heuristiques ont été réalisées afin d'avoir une possibilité de faire des comparaisons.

Nous terminons ce mémoire par une conclusion où nous indiquons les voies possibles pour les futures recherches.

# Chapitre 1 : Gestion des stocks et planification des chaînes logistiques

## 1.1 Introduction

Le but de la gestion de la chaîne logistique est d'être capable de mettre sur le marché, avant les concurrents, des produits de bonne qualité, de faible coût, et qui répondent aux désirs exprimés par les clients.

Dans ce chapitre nous nous intéressons à présenter des généralités sur la gestion des chaînes logistiques et précisément sur la gestion des stocks et la planification des chaînes logistiques sous incertitudes. Ces incertitudes peuvent être classées en deux types : sur la quantité et sur le temps. L'incertitude sur la quantité concerne par exemple la demande de clients et l'incertitude sur le temps concerne par exemple les délais d'approvisionnement en composants auprès des fournisseurs.

Sous ces incertitudes, la gestion des stocks permet donc de déterminer le niveau de stock de chaque article afin de réduire le coût de possession (ou de stockage), le coût de passation des commandes tout en respectant un niveau désiré de service des clients. L'objectif de la gestion des stocks est donc de trouver un compromis entre le niveau de stock (coût de stockage) et la satisfaction des clients (taux de service, minimisation de coût de rupture). En effet, si l'on s'intéresse à minimiser le coût de stockage sans se soucier de taux de service on risque à ne pas satisfaire les clients et donc de perdre certaines commandes et voir même perdre des clients non satisfaits. A l'inverse, avoir un niveau de stocks trop élevé conduit à un coût de stockage aussi trop élevé.

La planification des approvisionnements d'une chaîne logistique reprend l'idée que la satisfaction d'un client est le résultat de la mise en œuvre d'une succession de processus sans trop se préoccuper du périmètre juridique de l'entreprise, en remontant, si nécessaire, jusqu'à l'approvisionnement en matières premières (Giard, 2003). La planification des approvisionnements consiste donc à coordonner efficacement les différents maillons de la chaîne logistique afin d'offrir les produits en bonne quantité, au bon endroit et au bon moment, et de minimiser le coût global, tout en obtenant un niveau de service

suffisant pour tous les partenaires de la chaîne logistique. Les objectifs de la planification des approvisionnements varient selon chaque entreprise, ils peuvent être par exemple : améliorer le taux de service des clients, réduire les délais, réduire les retards, augmenter les profits, réduire les coûts, etc.

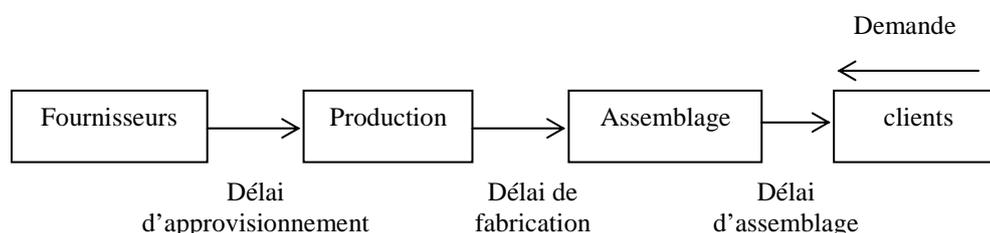
La planification des approvisionnements joue donc un rôle primordial afin de bien gérer la chaîne logistique en garantissant un bon compromis entre la minimisation de coût total (stockage, lancement, en-cours, ..., etc) et maximisation de niveau de service (réduire le rupture) pour le client final.

Une description générale des chaînes logistiques est faite dans la section suivante : nous présentons la définition des chaînes logistiques et ses différentes structures possibles. Ensuite, dans la section 1.3, les méthodes de gestion des stocks sont introduites à travers une présentation de quelques politiques classiques. Dans la section 1.4, nous nous intéressons à la planification de type *Material Requirement Planning* MRP. Dans la section 1.5, nous introduisons la problématique de la présente thèse. Enfin, la section 1.6 présente des conclusions.

## 1.2 Chaînes logistiques « Supply chain »

### 1.2.1 Définition

Une chaîne logistique est un réseau d'organisations qui contribuent aux différents processus et activités, à travers les interactions en amont et en aval, apportant une valeur ajoutée sous la forme de produits et de services pour les clients finaux. D'un point de vue conceptuel, une chaîne logistique peut être considérée comme une succession de processus d'approvisionnements, de fabrication, de distribution et de vente d'un produit, depuis le premier des fournisseurs jusqu'au client final (Mollet *et al.*, 2006). Une chaîne logistique est donc constituée de fournisseurs, de centres de production, d'entrepôts de stockage, de centres de distribution et de points de vente, le tout traversé par un flux physique qui transforme progressivement les matières premières et composants en produits finis. Une illustration de la chaîne logistique est donnée dans la figure 1.1.



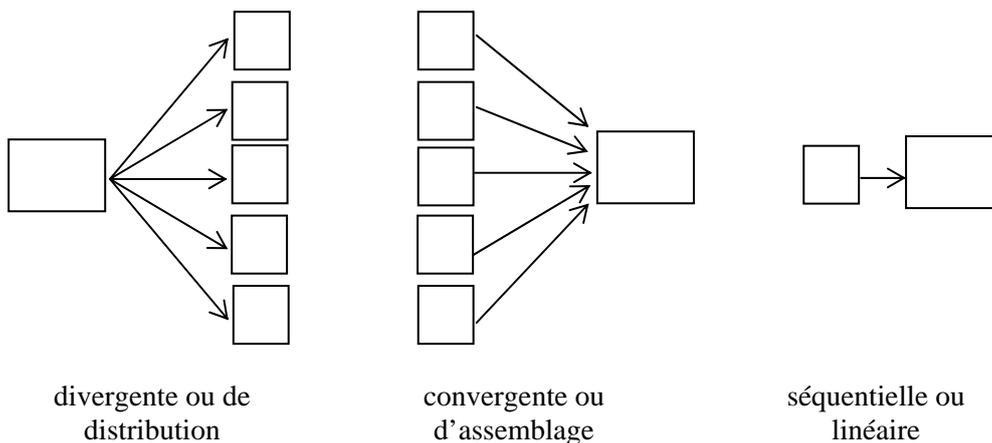
**Figure 1.1 :** Chaîne logistique

## 1.2.2 Structures possibles de la chaîne logistique

La structure d'une chaîne logistique dépend évidemment de sa nature et des objectifs souhaités lors de sa conception. Plusieurs architectures existent. Elles peuvent être classifiées comme suit (voir la figure 1.2) :

- Divergente ou de distribution : une chaîne est dite divergente si un fournisseur alimente plusieurs clients, plusieurs fournisseurs ou un réseau d'entreprises.
- Convergente ou d'assemblage : une chaîne est dite convergente si un client où une entreprise est alimentée par plusieurs fournisseurs. Cette structure est également présente dans les systèmes d'assemblage.
- Séquentielle ou linéaire : chaque entité de la chaîne alimente une seule autre entité en aval.

On peut aussi trouver plusieurs structures qui sont des combinaisons des celles-ci.



**Figure 1.2** : Structures possibles d'une chaîne logistique

Ces différentes structures des chaînes logistiques peuvent être à un seul niveau ou à plusieurs niveaux. Dans la figure suivante (voir la figure 1.3) nous présentons un exemple d'une chaîne logistique avec une structure linéaire à trois niveaux (voir 1.3 a) et un exemple de système d'assemblage à deux niveaux (voir 1.3 b).

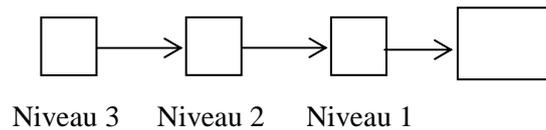


Figure 1.3 a : Chaîne logistique à structure linéaire à trois niveaux

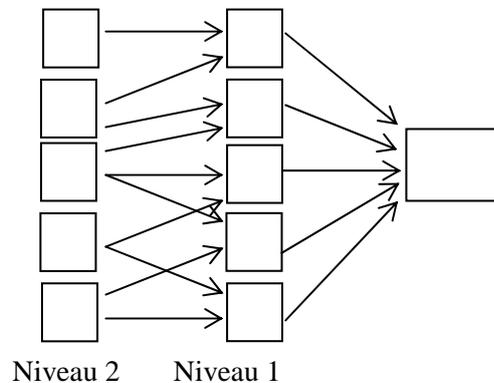


Figure 1.3 b : Système d'assemblage à deux niveaux

**Figure 1.3 :** Exemples des chaînes logistiques

Dans notre étude, dans le chapitre 3, nous intéressons à la planification et à la gestion des stocks des chaînes logistiques à structures linéaires multi-niveaux. Ensuite, les cas de gestion des stocks et de planification d'approvisionnements dans des systèmes d'assemblages à deux niveaux feront l'objet des chapitres 4 et 5.

### 1.3 La gestion des stocks

La chaîne logistique est composée d'un ensemble de processus et maillons et de relations entre eux visant à optimiser le déplacement des produits, dans l'espace et dans le temps, en vue de répondre plus efficacement aux exigences des clients et ce, au coût le plus bas.

Cette définition met en relief trois composantes fondamentales de la chaîne logistique : les maillons de la chaîne, les flux de matières et les flux d'information. La chaîne logistique est donc un réseau de partenaires (ou maillons) échangeant des matières et de l'information, dans le cadre d'activités menant ultimement à la livraison des produits (ou des services) aux clients finaux.

Une gestion saine de la chaîne logistique consiste à planifier stratégiquement ses opérations, à s'approvisionner en élaborant de bonnes approches de collaboration entre les partenaires, à produire efficacement et à distribuer en respectant les niveaux de services, grâce à des réseaux d'approvisionnements optimisés.

La gestion des stocks de la chaîne logistique implique tous les maillons de la chaîne en incluant le fournisseur du fournisseur en partant du client final. La gestion de ces relations d'affaires est cruciale au succès de la chaîne. L'accent devrait être mis à développer des relations durables et profitables entre tous ses maillons.

En plus, toutes les informations échangées à l'intérieur de chaque maillon doivent être partagées avec les autres maillons afin de maximiser l'efficacité de l'ensemble. Cela permettra aux maillons de la chaîne de pouvoir réagir à temps aux problèmes qui surviennent et de prendre les meilleures décisions possibles.

Une des caractéristiques fondamentales de gestion des stocks dans la chaîne logistique est que chacun des maillons de la chaîne logistique a, positivement ou négativement, un impact sur le reste de la chaîne. Ainsi, toute rupture de marchandises chez un des fournisseurs se répercutera jusqu'au client final et tout changement de la demande chez le client finale se répercutera jusqu'au fournisseur de la matière première.

Enfin, notons que la définition ci-haut de la chaîne logistique comporte trois sphères d'activités : l'approvisionnement, la production et la distribution des produits. L'importance d'une bonne gestion de la chaîne logistique n'est plus à démontrer. En fait, la seule question qui demeure est comment trouver les bons paramètres de la gestion des stocks (stocks de sécurité, délais de sécurité, règles de lot-sizing) à tous les niveaux afin de satisfaire le client final à la date souhaitée au moindre coût.

### **1.3.1 Les paramètres de la gestion des stocks**

Nous étudions quelques paramètres qui influent sur le coût de stock :

-Le prix unitaire d'achat : il est possible d'obtenir des remises quantitatives et ainsi, le prix unitaire d'achat est réduit par le système de tarif à seuil.

-Le délai d'approvisionnement : il représente le temps entre le moment où l'ordre doit être livré (la due date) et la date de lancement de l'ordre. Il a une influence sur la gestion des stocks que nous montrons dans les chapitres suivants.

-Le coût de lancement et le coût de possession d'une commande : l'objectif d'une bonne gestion des stocks est de trouver le nombre optimal de lancements en tenant compte des délais entre deux lancements, le coût de lancement d'une commande et son coût de possession afin de réduire le coût total de lancement sur un horizon d'une année, par exemple.

-Les politiques de gestion des stocks : une fois les paramètres qui interviennent dans le calcul du coût de stockage sont connus, nous pouvons déterminer les politiques de gestion des stocks qui minimisent le coût total en tenant compte de la satisfaction des clients.

### **1.3.2 Origine et nature des stocks dans une chaîne logistique**

Dès qu'il y a un décalage horaire ou en quantité entre la production et la demande, des stocks ou des ruptures d'approvisionnement se produisent. Pour éviter ces dernières, des stocks complémentaires sont également constitués. Les stocks se trouvent ainsi à tous les niveaux de la chaîne logistique. Ces stocks sont sous forme de produits finis ou de produits semi-finis en cours de fabrication que nous appelons « en-cours ». Ces derniers peuvent représenter des stocks importants ; souvent plus de 95% de temps total passé dans les ateliers les pièces restent en stocks (Mollet *et al.*, 2006). Les stocks dépendent également de la taille des lots lancés en fabrication, de la planification, de l'ordonnancement et de l'incertitude de la demande et des délais. Dans l'industrie il y a des centaines de milliers de références des pièces chez les fournisseurs, en usines et dans les réseaux d'après-vente réparties chacune sur plusieurs point de stockage. Le recours à des méthodes avancées de gestion des stocks est donc nécessaire pour traiter des stocks dont la valeur immobilisée représente souvent plusieurs mois de chiffre d'affaire.

Les stocks étant indispensables au fonctionnement de l'entreprise et à son service client, il convient de les dimensionner aux meilleurs niveaux. Nous allons maintenant examiner les politiques classiques de gestion des stocks qui ont cet objectif.

### **1.3.3 Les politiques classiques de gestion des stocks**

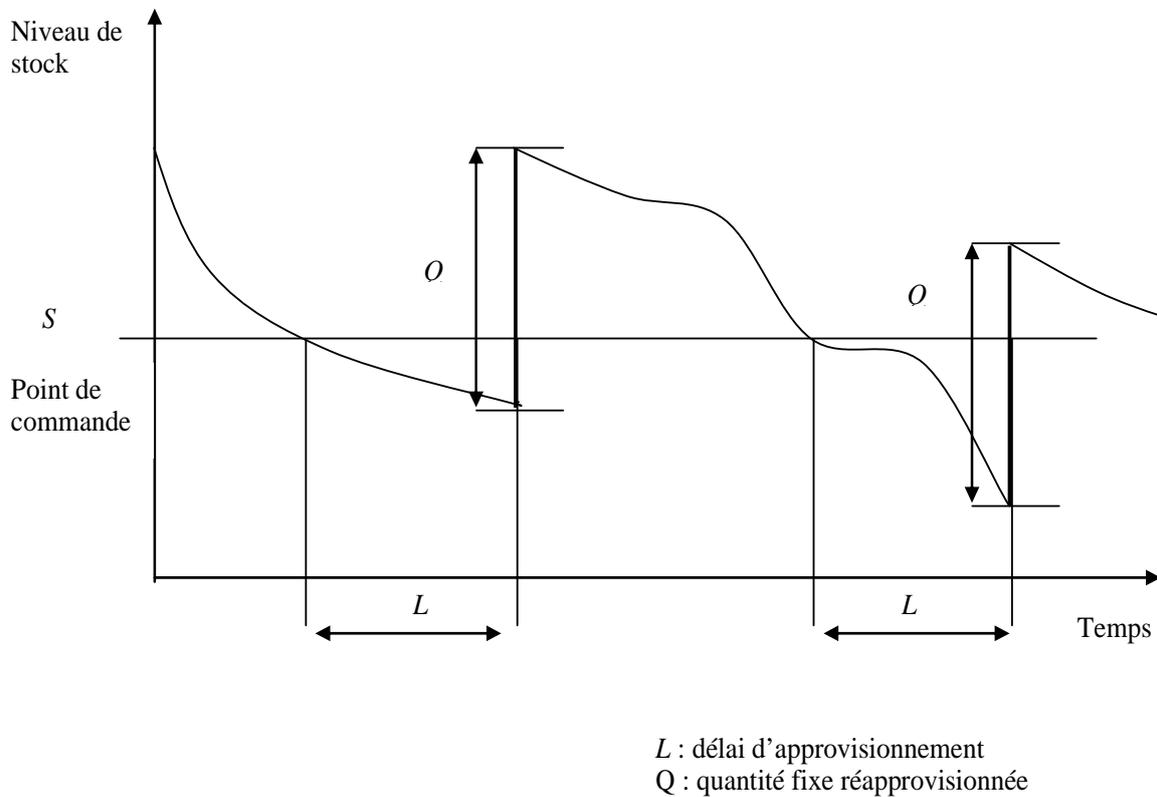
Une politique de gestion des stocks peut se résumer à un type de décisions d'approvisionnement d'un stock pour satisfaire une demande. Pour une décision d'approvisionnement, il faut déterminer conjointement une date de commande et une quantité à commander. Les politiques classiques de gestion des stocks se divisent en deux catégories suivant que l'on met en avant la période ou la quantité. Cela donne d'une part, des politiques de gestion des stocks à période fixe et quantité d'approvisionnement variable et des politiques à quantité fixe et à période de réapprovisionnement variable.

Pour chacune de ces politiques, il existe une variété extrêmement importante des modèles suivant les hypothèses retenues sur les éléments de la politique de la gestion. Nous ne donnons ici qu'un exemple de politique à quantité fixe.

#### **1.3.3.1 Un exemple de politique à quantité fixe : le point de commande**

C'est une méthode très simple et très répandue pour les articles peu chers et de consommation régulière. Le principe consiste à définir un seuil  $S$ . Dès que le niveau du stock atteindra le seuil  $S$ , on déclenchera l'approvisionnement d'une quantité  $Q$ . C'est ce seuil qui est appelé le point de commande car il déclenche le réapprovisionnement (ou la commande). Cette politique de la gestion des stocks demande à définir deux paramètres, d'une part, le seuil (le point de commande) et d'autre part, la quantité de réapprovisionnement. Le point de commande a pour but d'éviter les ruptures, c'est une

valeur de stock qui correspond à la quantité qui sera demandée ou consommée pendant le délai de réapprovisionnement. Dans le cas où la demande ou le délai d'approvisionnement serait aléatoire, lors de la détermination du point de commande cette quantité est généralement majorée en utilisant un délai de sécurité. La quantité de réapprovisionnement est aussi appelée taille de lot, voir la figure 1.4 (Mollet *et al.*, 2006).

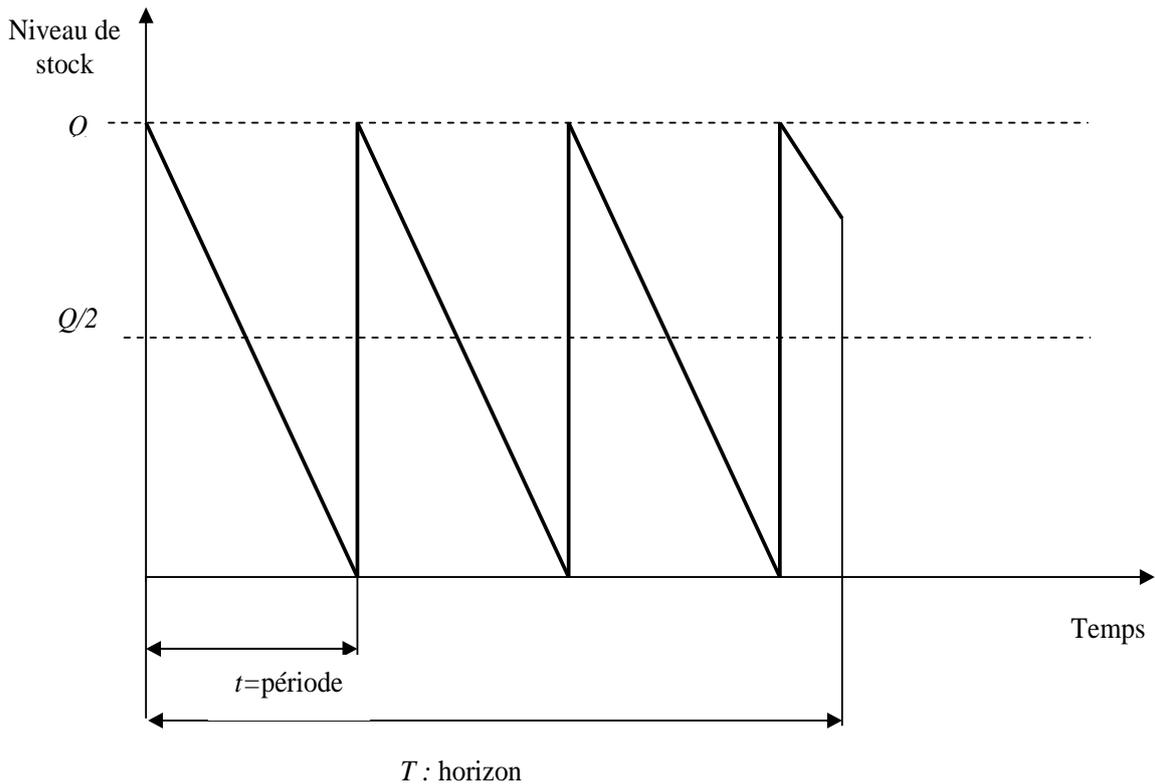


**Figure 1.4** : Politique d'approvisionnement à point de commande

### 1.3.3.2 Quantité économique EOQ

La quantité de commande peut être déterminée en utilisant la formule du lot économique EOQ (Economic Order Quantity). C'est le modèle de dimensionnement de stocks le plus courant.

Pour le modèle à quantité économique EOQ, la consommation est supposée régulière et le délai de livraison est connu. Le stock moyen  $q$  vaut la moitié de la quantité commandée  $Q/2$ . Optimiser le stock revient alors à optimiser la quantité commandée. Cette quantité  $Q$  est appelée quantité économique. Ce modèle est représenté dans la figure 1.5.



**Figure 1.5 :** Politique d'approvisionnement à quantité économique

Harris (1913) a montré qu'il existe une quantité économique  $Q^*$ , qui permet de minimiser le coût total composé du coût de lancement  $Cl$  et du coût de stockage  $h$ , elle est donnée par la formule suivante :

$$Q^* = \sqrt{2L \frac{Cl}{h}} \quad (1.1)$$

Dans les paragraphes suivants, nous présentons un autre modèle celui de Newsboy avec deux modifications : lorsque le délai d'approvisionnement nul et la demande est aléatoire et lorsque la demande est connue et déterministe et le délai d'approvisionnement est non nul et aléatoire.

### 1.3.3.3 Modèle de Newsboy avec demande aléatoire et délai d'approvisionnement nul

La gestion des stocks peut également être faite avec le modèle de Newsboy qui tient en compte du coût de possession  $h$  et du coût de rupture  $b$ . Dans ce modèle, la demande est considérée comme une variable aléatoire avec une distribution connue (Porteus, 1990 ; Dupont 1998).

Pour expliquer le modèle de Newsboy avec une demande aléatoire, nous représentons un exemple de (Dupont, 1998). Un marchand de journaux achète chaque jour une quantité fixe d'un certain quotidien. Si la demande est supérieure à son stock, il constate un manque à gagner  $b$ . Tout exemplaire non vendu engendre une perte  $h$ . L'objectif est d'augmenter le gain total et donc de trouver la quantité optimale à avoir en stock.

Soit  $p_x$  la probabilité que la demande  $D$  soit égale à  $x$  articles, et  $S$  le niveau de stock (approvisionnement) :

1) si  $x < S$ , alors il reste  $S-x$  invendus, le coût de possession est  $h(S-x)$  ;

2) si  $x > S$ , alors il manque  $x-S$  articles, le coût est de  $b(x-S)$ .

Le coût total est donc :

$$C(S) = h \sum_{x=0}^{S-1} (S-x)p_x + b \sum_{x>S} (x-S)p_x$$

La solution optimale  $S^*$  est telle que :

$$F(S^* - 1) \leq \frac{b}{b+h} \leq F(S^*)$$

où  $F(.)$  est la fonction de répartition de la variable aléatoire  $x$ .

#### 1.3.3.4 Modèle de Newsboy avec demande connue et délai d'approvisionnement aléatoire

Nous pouvons également utilisé le modèle de Newsboy pour des problèmes d'approvisionnement. Nous allons montrer ici comment.

L'un des aspects les plus délicats dans la planification des approvisionnements est la détermination des délais prévus d'approvisionnement. Le délai d'approvisionnement prévu (planned lead time) représente le temps entre le moment où l'ordre doit être livré (la due date) et la date de lancement de l'ordre (planned order release date). La détermination de délai d'approvisionnement est très importante, une erreur ici peut affecter sérieusement le bon fonctionnement des systèmes de planification des approvisionnements.

Soit  $x$  le délai d'approvisionnement prévu, nous avons alors :

$$x = T - X \tag{1.2}$$

où,

$T$  : représente la date voulue de livraison,

$X$  : représente la date de lancement d'ordre à un fournisseur.

Ces paramètres sont utilisés, entre autre, par les systèmes de type MRP où les dates de livraison de produits finis sont données par le Plan Directeur de Production (PDP), c.-à-d. « Master Production Schedule » (MPS) et l'équation (1.2) est ensuite utilisée d'une manière itérative pour déterminer, niveau par niveau, la date de lancement d'ordre pour tous les composants.

L'ordre mis en place peut prendre plus au moins de temps que ce qui a été alloué (le temps d'approvisionnement prévu). Le délai d'approvisionnement réel  $L$  est donc exprimé comme suit :

$$L = C - X \quad (1.3)$$

où,

$C$  : représente la date réelle d'arrivé des composants commandés,

$X$  : représente la date de lancement d'ordre à un fournisseur.

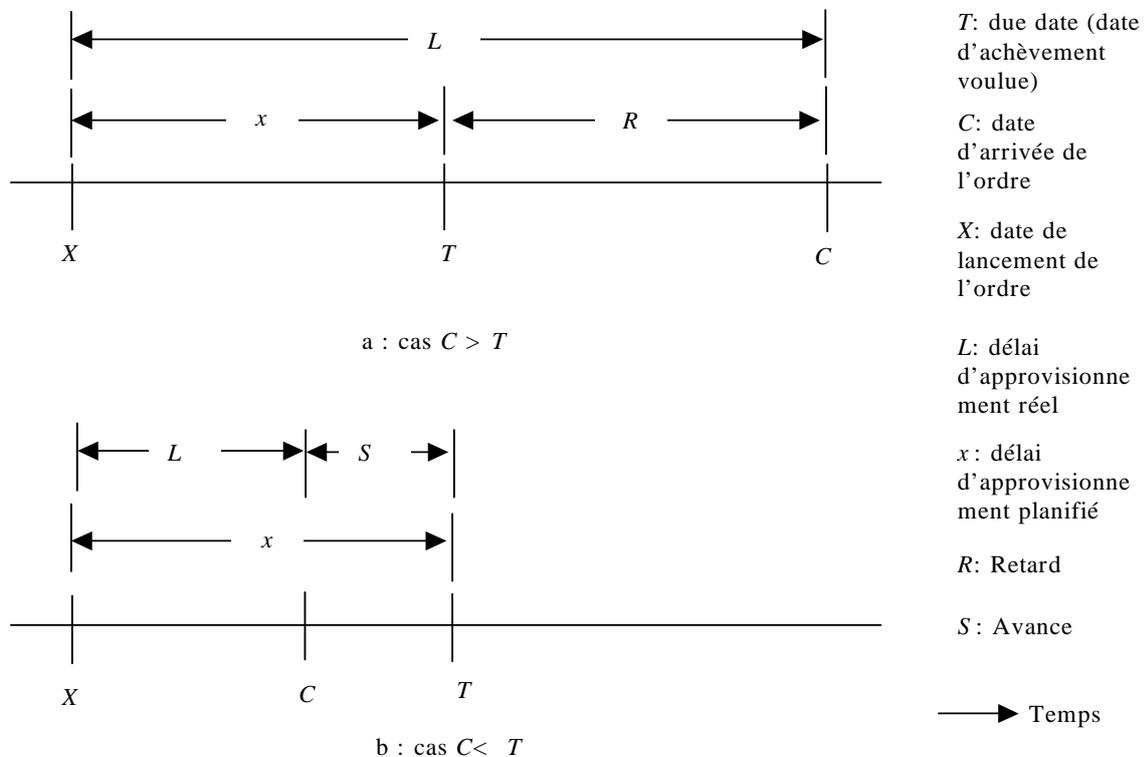
Si le délai d'approvisionnement réel  $L$  est supérieur au délai d'approvisionnement planifié  $L - x > 0$  (voir la figure 1.6a) alors il y a un retard. Dans le cas où cette différence est négative  $L - x < 0$  (voir la figure 1.6b), le produit est en avance. Le retard  $R$  (rupture) et l'avance  $S$  (stock) sont donc modélisés comme suit :

$$R = (L - x)^+ \quad (1.4)$$

$$S = (x - L)^+ \quad (1.5)$$

où  $(.)^+$  est  $\max(.,0)$

La figure 1.6 présente la relation entre délai d'approvisionnement prévu ou planifié et le délai d'approvisionnement réel.



**Figure 1.6 :** Illustration de la relation entre les délais d'approvisionnement planifié et réel

Weeks (1981) a montré que le modèle de Newsboy peut également être utilisé pour déterminer le délai d'approvisionnement planifié pour le cas de la demande connue et du délai d'approvisionnement non nul et aléatoire.

L'objectif est de réduire le coût total moyen composé de coût de stockage et de coût de rupture. Le problème donc est de trouver le délai d'approvisionnement planifié qui permet de réduire ce coût total.

Le coût total est donc :

$$C(x) = h \sum_{L=0}^{x-1} (x-L)p_L + b \sum_{L>x} (L-x)p_L$$

où  $p_L$  est la probabilité que le délai d'approvisionnement réel est égal à  $L$ .

La solution optimale est telle que :

$$F(x^* - 1) \leq \frac{b}{b+h} \leq F(x^*)$$

où  $F(.)$  est la fonction de répartition de la variable aléatoire  $L$ .

## 1.4 La politique de planification de type MRP

La planification d'approvisionnements efficace est une fonction très importante pour la gestion des entreprises industrielles. En effet, une mauvaise politique de gestion des stocks conduit soit à des stocks inutiles ou à des ruptures. Les stocks sont créés à différents niveaux (des matières premières aux produits finis), coûtent de l'argent et immobilisent des fonds. Une rupture de stock conduit à une pénalité due aux commandes insatisfaites.

Les méthodes de planification de type MRP sont les méthodes les plus connues dans le contexte industriel (Axsäter, 2006). Les systèmes MRP sont acceptés facilement par les industriels, la majorité des décideurs industriels sont familiers avec eux à travers tous les systèmes existants de gestion informatique de la production. Les techniques MRP disposent d'un système d'information bien développé et ont fait ses preuves au fil du temps.

Toutefois, MRP est basée sur l'hypothèse que la demande et les délais d'approvisionnement sont connus. Cependant, dans le monde industriel, on constate très souvent que les délais d'approvisionnement varient souvent d'une manière aléatoire. Les délais d'approvisionnements des composants finis sont rarement prévisibles d'une manière fiable. En effet, il y a certains facteurs aléatoires tels que les pannes de machines, le retard de transport, etc. Par conséquent, les hypothèses déterministes intégrées dans les systèmes MRP concernant le temps d'approvisionnements sont souvent trop limitées.

Comme a été indiqué dans plusieurs articles, différentes sources d'incertitudes peuvent exister le long de la chaîne logistique. Pour éviter les conséquences de ces incertitudes, les entreprises utilisent les stocks de sécurité ou des délais de sécurité, les deux provoques des stocks complémentaires qui sont assez chers. Par conséquent, il est souhaitable de développer des méthodes de planification des approvisionnements qui tiennent compte des propriétés stochastiques des délais d'approvisionnements (Maloni et Benton, 1997).

Dans l'approche MRP, une distinction importante est faite entre la demande pour le produit final, c'est-à-dire la demande indépendante, et la demande d'un de ses composants, c'est-à-dire la demande dépendante. La demande indépendante est connue ou prévue par les méthodes qui sont élaborées dans le cadre de "la prévision des ventes". Les demandes dépendantes peuvent être calculées à partir de la demande indépendante à l'aide de la nomenclature du produit finis correspondant et les délais d'approvisionnement planifiés.

Sous la logique MRP, le temps est dirigé en intervalles discrets appelés « *time buckets* ». Le délai d'approvisionnement est égal au nombre d'intervalles écoulés entre la date de lancement de l'ordre et sa date de livraison. La taille du lot est la quantité des articles commandés.

La méthode MRP est basée sur le calcul déterministe : toutes les commandes des composants sont lancées au plus tard, donc le coût total sera automatiquement le plus petit possible. Mais, s'il existe des facteurs aléatoires, le sens de l'expression «au plus tard» est incertain. Dans ce cas, pour chaque paramètre de la méthode MRP, nous pouvons avoir une probabilité de stockage des composants et une probabilité de rupture en produit finis. Par conséquent, il faut utiliser le paramétrage de la méthode MRP, c.-à-d. le choix des valeurs optimales pour ses paramètres minimisant les stockages et/ou les ruptures. Ce problème est appelé le paramétrage des systèmes MRP sous incertitudes.

#### **1.4.1 Principe de la méthode MRP**

Le principe de la méthode MRP est de déterminer sur un horizon donné, les sorties prévisionnelles ou les ordres de fabrications à lancer dans l'atelier. Les besoins en produits finis sont donnés par le Plan Directeur de Production (PDP), et ceux en composant en sont déduits par éclatement de la nomenclature (Dolgui et Prodhon., 2007). Un exemple de nomenclature est présenté dans la figure 1.7.

Un certain nombre de données sont donc nécessaires comme les nomenclatures des produits. Il est également nécessaire de connaître l'état des stocks, les réceptions prévues, les besoins bruts, les fournisseurs, les tailles de lot, les délais d'approvisionnements ou d'obtention. Ces derniers sont utiles pour savoir quand est-ce qu'il faut lancer une commande. Le délai d'approvisionnement correspond au temps qu'il faut entre le moment où l'on décide de commander et le moment de la réception de la commande. On peut également y ajouter une marge de sécurité appelée délai de sécurité jouant un rôle similaire au stock de sécurité.

Une fois ces données sont obtenues, on calcule les différents besoins en composants. On les obtient par éclatement de la nomenclature déterminant la quantité de chaque composant nécessaire pour assurer la fabrication voulue de produits finis. On ajoute éventuellement des besoins indépendants correspondant par exemple aux pièces de rechanges, si besoin.

Il reste donc à déterminer quand est-ce qu'il faut commander. Il faut alors considérer le temps que cela prend pour obtenir ces composants. Ensuite, on les commande au plus tard en tenant compte de l'état des stocks et des délais d'approvisionnements.

Si on commande uniquement les quantités nécessaires pour chaque date, il s'agit de la politique Lot-pour-Lot (ou Lot-For-Lot, LFL). Cependant, il est parfois préférable de grouper des lots en tenant compte des frais de passation de commande et de possession des stocks. Il existe différentes techniques de regroupement appelées aussi lotissements ou lot-sizing (Vollmann *et al.*, 1997).

Prenons un exemple de lot pour lot politique (Dolgui et Prodhon., 2007) :

$I(i)$  Stock disponible à la période  $i$ ,

$N(i)$  Besoin net de la période  $i$ ,

$G(i)$  Besoin brut pour la période  $i$ ,

$Q(i)$  La fabrication ou l'achat à la période  $i$ ,

$\Delta\tau$  Délai d'approvisionnement.

Le stock disponible en première période  $I(1)$  est donné, puis pour les périodes suivantes, il est déduit du besoin net de la période précédente :

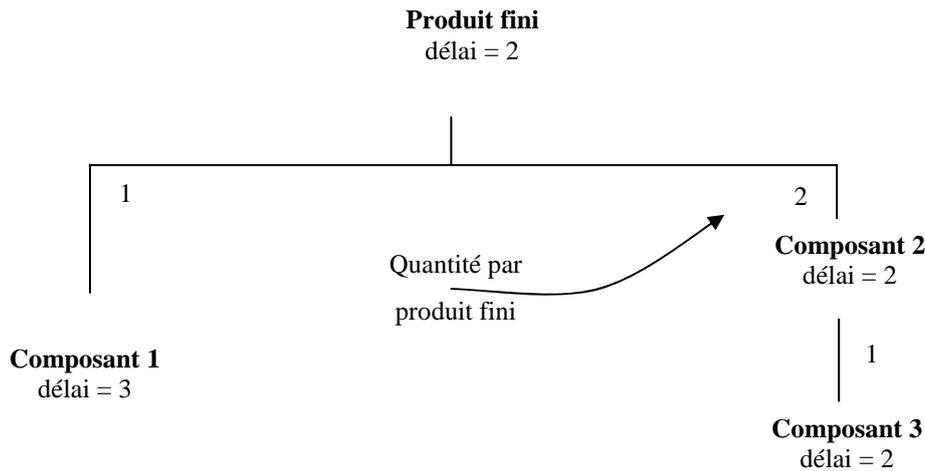
$$I(i) = \max\{0, -N(i-1)\}, \quad (1.6)$$

Le besoin net à la période  $i$  est égal au (besoin brut – stock disponible), il est obtenu comme suit :

$$N(i) = G(i) - I(i), \quad (1.7)$$

L'ordre de fabrication ou d'achat :

$$Q(i) = \max\{0, N(i + \Delta\tau)\}, \quad (1.8)$$



**Figure 1.7 :** Nomenclature de produit fini

Produit Fini	période	1	2	3	4	5	6	7	8	9	10
		besoin brut (PDP)	0	0	0	50	10	40	20	30	50
Niveau 0	stock disponible	20	20	20	20	0	0	0	0	0	0
Délai = 2	besoin net	-20	20	-20	30	10	40	20	30	50	60
	fabrication/achat	0	30	10	40	20	30	50	60	0	0

Quantité = 1

Composant1	période	1	2	3	4	5	6	7	8		
		besoin brut	0	30	10	40	20	30	50	60	
Niveau 1	stock disponible	100	100	70	60	20	0	0	0		
Délai = 3	besoin net	-100	-70	-60	-20	0	30	50	60		
Quantité = 1	fabrication/achat	0	0	30	50	60	0	0	0		

Délai = 3

Composant2	période	1	2	3	4	5	6	7	8		
		besoin brut	0	60	20	80	40	60	100	120	
Niveau 1	stock disponible	140	140	80	60	0	0	0	0		
Délai = 2	besoin net	-140	-80	-60	20	40	60	100	120		
Quantité = 2	fabrication/achat	0	20	40	60	100	120	0	0		

Composant3	période	1	2	3	4	5	6	7	8		
		besoin brut	0	20	40	60	100	120			
Niveau 2	stock disponible	50	50	30	0	0	0				
Délai = 2	besoin net	-50	-30	10	60	100	120				
Quantité = 1	fabrication/achat	10	60	100	120	0	0				

**Figure 1.8 :** Plan Directeur de Production (PDP)

### 1.4.2 Les incertitudes sous MRP

La méthode MRP se situe volontairement en univers certain. Cette hypothèse de travail n'est pas toujours réaliste. La demande et les délais d'approvisionnements sont souvent aléatoires (panne machine, casse d'un outil, absentéisme, retard d'un fournisseur, problèmes de qualité, commandes

urgentes, rendement aléatoire ...). Il faut alors trouver les paramètres qui conviennent pour aboutir à des résultats qui répondent le mieux aux besoins des clients tout en limitant les coûts engendrés.

Deux sources d'incertitude touchent principalement la méthode MRP (Nahmias, 1997; Vollmann *et al.*, 1997) : la demande et les approvisionnements (voir la figure 1.8). On peut encore diviser ces incertitudes en deux catégories : quantité et temps. La première signifie que les quantités fournies sont différentes des quantités réelles demandées. La deuxième concerne les dates de demandes ou de réceptions qui ne sont pas respectées.

Enns (2002) a étudié l'effet de l'erreur de la prévision sur la performance de MRP et comment l'incertitude de la demande peut être mieux traitée. L'auteur s'est intéressé à l'impact de l'augmentation de stocks et de délais d'approvisionnements sur le niveau de service. A travers la simulation, il a montré que le niveau de service augmente lorsque le ratio (prévision/demande) augmente. Il a également montré qu'en augmentant les stocks de sécurité et les délais d'approvisionnements le niveau de service augmente aussi.

Inderfurth (2007) a étudié la performance des systèmes MRP et ses paramétrer sous les incertitudes de la demande et du rendement. La demande était supposée stationnaire, le délai d'approvisionnement égal à zéro et le rendement était proportionnel à la quantité à commander. L'auteur a présenté une expression analytique de stock de sécurité pour les lois connues (uniforme, normale) de la demande et du rendement quand la politique d'approvisionnement est la politique Lot pour Lot.

Produit Fini	Période	1	2	3	4	5	
Niveau 0 Délai = 2 +/- 1	Besoin brut (PDP)	0	0	20	15	0	↑ Incertitude en quantité pour la demande ↓
	stock disponible	20	20	20	0	0	
	besoin net	-20	-20	0	15	0	↑ Incertitude en quantité pour les approvisionnements ↓
	fabrication/achat		15				

↔ Incertitude en temps pour le délai d'approvisionnement ↔  
 ↔ Incertitude en temps pour la demande ↔

**Figure 1.9 :** Incertitudes des données pour la méthode MRP

L'incertitude sur la demande signifie que la demande n'est pas exactement connue à l'avance et donc les quantités prévues peuvent être différentes de la demande réelle. Le délai d'approvisionnement incertain signifie que le délai réel peut être différent du délai planifié, donc une commande planifiée pour une période peut ne pas arriver à la date appropriée. L'incertitude de rendement signifie que la quantité commandée ou fabriquée est différente de la quantité produite ou fournie.

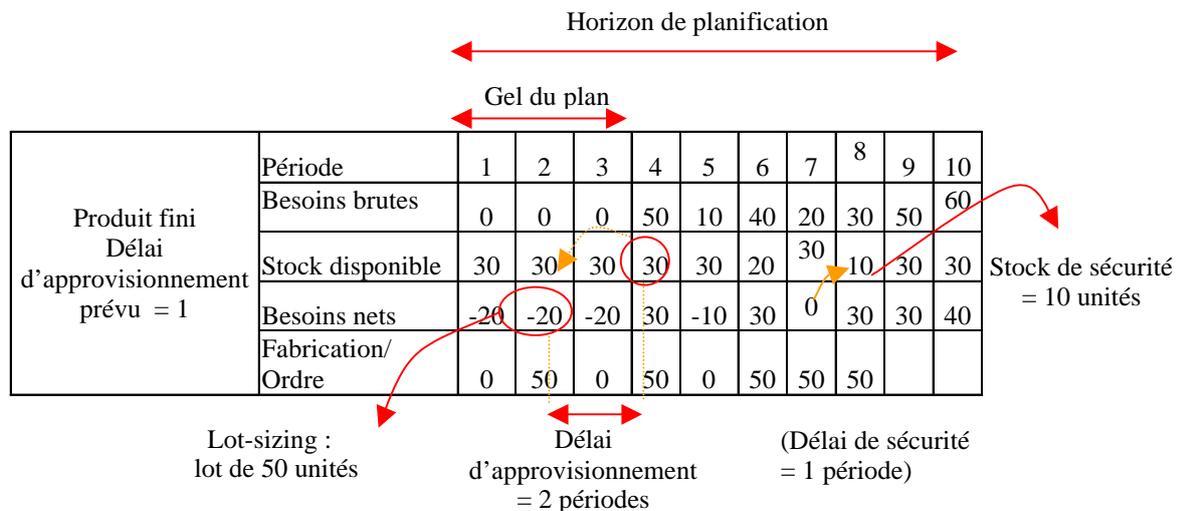
Ces incertitudes peuvent provoquer une replanification. Par exemple, on peut avoir prévu une demande de  $D$  unités et finalement n'avoir que  $D - \Delta$  unités, il en reste alors  $\Delta$  unités manquantes et

pour ne pas faire de surstock, il faut réduire la production pour la période suivante. Mais on peut également avoir le phénomène inverse et devoir changer le plan directeur de production d'urgence pour pouvoir satisfaire la demande plus grande que prévue.

Plus généralement, tout changement dans les données ou des paramètres du système MRP provoque des instabilités qui ont des effets négatifs sur l'ensemble du système (coût de stockage et coût de rupture, difficultés de gestion, etc.). Il est donc nécessaire de trouver les paramètres optimaux qui permettent de minimiser les effets négatifs des incertitudes des données.

Les paramètres qui peuvent être modifiés dans le système MRP en cas d'incertitudes sont de différentes natures (voir la figure 1.10) :

- stocks de sécurité
- délais de sécurité / temps de cycle planifiés
- taille des lots
- gel d'une partie de PDP
- choix du horizon de planification
- ...



**Figure 1.10 : Paramètres de MRP**

### 1.4.3 Stocks et délais de sécurité

La prise en compte des variations possibles des besoins brutes peut s'effectuer par le biais d'un stock de sécurité (*safety stock*). La particularité de ce stock de sécurité est d'être d'usage exceptionnel. Il n'a pas pour but d'être écoulé entièrement entre deux livraisons, mais de pallier aux aléas de quantité. L'impact de ce stock est double : il réduit le risque de rupture et augmente les coûts de possession (coût de stockage). Il faut donc l'ajuster en fonction de deux objectifs suivants :

- Minimiser les coûts de rupture et de stockage
- Minimiser le coût de stockage tout en satisfaisant le niveau de service souhaité par les clients.

Notons qu'il est possible selon Plenert (1999) de réduire voir même de supprimer la plupart des stocks de sécurité en instaurant des capacités de production de sécurité.

Comme a été décrit précédemment, la méthode MRP se base sur une vision déterministe de la demande et des délais dans les systèmes de production. Pour faire face aux différents aléas, la notion du stock de sécurité a été introduite. Graves (1988) montre l'importance d'utilisation des stocks de sécurité et donne une analyse des méthodes le plus utilisées pour les dimensionner. Simpson (1958), Hansmann (1959) et Miller (1979) ont proposé des heuristiques qui consistent à fixer des seuils des aléas et de calculer les stocks de sécurité qui permet la satisfaction de la demande si les aléas ne dépassent pas les seuils fixés.

Giard (2003) indique qu'une des principales difficultés est la détermination des niveaux des stocks de sécurité (et leurs variations).

En ce qui concerne le délai de sécurité (*safety lead time*), il fonctionne sur le même principe, mais au lieu d'agir sur les quantités, on agit sur le temps. En utilisant les délais de sécurité, le délai d'approvisionnement planifié est déterminé en faisant la somme du délai d'approvisionnement prévisionnel et du délai de sécurité. Par exemple, on peut prendre un délai de sécurité égal à  $k$  fois l'écart-type du délai prévisionnel (Melnyk et Piper, 1981).

(Whybark et Williams, 1976) affirment que des stocks de sécurités sont à utiliser quand des incertitudes portent sur les quantités, et des délais de sécurités quand il y a des aléas dans les délais, afin de minimiser les coûts de stockage et d'assurer un bon niveau de service dans un système MRP.

Néanmoins, ce n'est pas la conclusion de Grasso et Taylor (1984), qui préfèrent les stocks de sécurité dans tous les cas. Ils se sont intéressés spécifiquement aux incertitudes sur les délais et ont travaillé sur quatre facteurs (variabilité des délais, stocks et délais de sécurité, technique de lot-sizing, valeur des pénalités des retards et coûts de possession) qui jouent un rôle sur le coût total des systèmes MRP. Leurs expérimentations concernaient trois produits de nomenclature à cinq niveaux chacun. Ils concluent de leur étude que ces quatre facteurs définissent les coûts de gestion MRP.

Quant à De Bodt et Van Wassenhove (1983), ils nous disent que les solutions de type stocks de sécurité ne sont appropriées que quand la variabilité de la demande est faible, et le temps entre les commandes est petit.

On peut combiner des stocks de sécurité et des délais de sécurité. Mais avant d'opter pour de telles techniques, il faut regarder s'il n'est pas possible de réduire les incertitudes par exemple en effectuant une meilleure prévision de la demande et des délais. On peut aussi, entre autre, appliquer quelques

techniques du Juste-à-Temps pour réduire les délais de fabrication, améliorer la qualité, c'est-à-dire diminuer les incertitudes.

#### **1.4.4 Méthodes de lotissement (lot-sizing)**

La méthode de lotissement la plus simple est Lot-pour-Lot (ou Lot-For-Lot, LFL). Quand on applique Lot-pour-Lot, il n'y a pas de regroupement des besoins de différentes périodes. On commande à chaque période selon les besoins nets d'une période plus lointaine (en tenant compte de délais d'approvisionnement) même lorsqu'il s'agit d'une demande de taille très faible. C'est la méthode qu'on utilise quand les coûts de lancement sont négligeables, son avantage est qu'elle minimise le stockage.

Toutefois, il est parfois plus judicieux de regrouper les besoins de plusieurs périodes, au lieu de travailler en Lot-pour-Lot. Si l'on rassemble les besoins de plusieurs périodes en créant des lots de taille plus importante les coûts de passation seront plus réduits (les coûts de lancement des ordres, éventuellement les coûts de transport,...), mais en cas de coût de possession élevé par rapport au coût de commande ou quand le stockage pose problème, le Lot-pour-Lot n'est pas si mauvais. De ce fait, souvent, lorsque l'on est en présence de produits multi-niveau, le regroupement est utilisé pour les matières premières et quelques articles intermédiaires, et la politique Lot-pour-Lot pour le reste.

Nous allons maintenant étudier différentes techniques de lotissement couramment utilisées.

Comme nous l'avons déjà indiqué, une des techniques les plus connues est la quantité économique de commande (EOQ), introduite par Harris en 1913. Cette méthode calcule une quantité à commander par la formule de Wilson. Cette technique consiste donc à chercher la quantité économique de commande à passer périodiquement. Cependant la méthode EOQ se base sur des hypothèses restrictives de demande certaine et distribuée uniformément tout au long de l'année et du délai de réapprovisionnement aussi certain.

L'algorithme Wagner-Within (1958) est une procédure qui détermine le coût minimal de commande pour une demande déterministe dynamique sans contrainte de capacité. Sous les conditions de demande dynamique, rupture non permise, stock initial nul, coûts linéaires de possession et coûts fixes de commande, il est optimal. Il analyse toutes les possibilités de commande pour répondre à la demande en utilisant une programmation dynamique et ainsi choisit la moins coûteuse. C'est-à-dire il cherche à déterminer la quantité cumulée des besoins de chaque période. Zangwill (1966) a généralisé l'algorithme de Wagner-Within en permettant les ruptures de telle sorte que les demandes non satisfaites soient différées à la période suivante.

L'algorithme de Wagner-Within ayant le temps de calcul important pour des problèmes de taille réelle (Jeunet et Jonard, 2000), de nombreuses heuristiques ont été développées comme les trois suivantes.

L'idée de l'heuristique de Silver et Meal (1973) est de couvrir  $p$  périodes avec un ordre d'approvisionnement, cet algorithme cherche la valeur  $p$  qui minimise le coût moyen de stockage par période. L'heuristique de Silver et Meal est souvent plus performante que l'algorithme de Wagner-Within en cas d'incertitudes.

La méthode du moindre coût unitaire, ou Least Unit Cost (LUC), est une procédure qui évalue les différentes quantités à commander en accumulant les besoins des périodes consécutives jusqu'à ce que le coût commence à augmenter.

Toujours sur le même principe, Matteis en 1968 introduit une technique économique de lotissement le Part Period Algorithm (PPA). Il divise les coûts de lancement (ou d'ordre d'approvisionnement) par les coûts de stockage par unité de composant et par unité de temps. Cette valeur est utilisée pour déterminer la taille de lot. L'auteur a présenté une étude comparative, entre son algorithme, l'algorithme de Wagner et Within et l'algorithme « Least Unit Cost ». Les résultats montrent que l'algorithme de Wagner et Within représente toujours la solution à coût minimum et l'algorithme Least Unit Cost à coût maximum. Mais l'inconvénient de l'algorithme de Wagner et Within par rapport aux deux autres algorithmes est qu'il est moins performant lorsque la demande peut varier. PPA est remarquablement stable pour les différentes variations de la demande avec une solution à coût faible.

Tous ces modèles sont à capacité infinie, mais il en existe d'autres avec capacité finie. Nous pouvons citer celui qui généralise l'algorithme de Wagner et Within, et dont le principe est de borner les quantités produites pour chaque période (voir état de l'art dans Lee et Nahmias, 1993). On peut trouver aussi par exemple l'Economic Lot Scheduling Problem introduit par Rogers en 1958 pour traiter des problèmes avec demande stationnaire.

Les logiciels disponibles n'offrent que quelques règles de calcul de lotissement. Parmi ces règles, on retrouve bien sûr celle du Lot-pour-Lot, de la quantité économique de Wilson, mais aussi parfois des algorithmes plus sophistiqués comme l'algorithme de Wagner et Within ou l'heuristique de Silver et Meal. Cependant, ces derniers requièrent du temps de calcul plus important. Afin de faire face au problème de temps de calcul, on peut regrouper les produits par famille en se basant sur les critères composants et outillages, pour ne travailler qu'avec ces familles (Giard, 2003).

Sachant qu'une décision de lotissement à un niveau se répercute à tous les niveaux descendants. Il est donc difficile de trouver un regroupement qui soit optimal sur l'ensemble des niveaux d'une nomenclature. Une taille de lot optimale pour un article peut engendrer des pertes pour les autres composants dépendant de lui. Plenert (1999) nous conseil d'appliquer le Lot-pour-Lot à tous les produits des classes  $A$  et la plupart des ceux de classe  $B$  selon la classification de Pareto. Ceci dit dans certains cas spécifiques, comme le démontre Ho et Lau (1994) avec des délais incertains, les règles de Silver et Meal donnent de meilleurs résultats.

## 1.5 Problématique de la thèse

La gestion des stocks et la planification des chaînes logistiques en présence des aléas dans un environnement MRP est une pratique courante. En effet, les techniques de planification de type MRP sont acceptées facilement par les décideurs industriels qui sont familiarisés avec elles à travers les logiciels de gestion de production assistée par ordinateur (GPAO).

La présente thèse de doctorat s'intéresse à la problématique de paramétrage de MRP en présence d'aléas de délais d'approvisionnement. Nous essayons de trouver des nouvelles méthodes de planification en présence d'aléas et nous montrons la possibilité de leur utilisation pour le paramétrage de MRP.

Le plus souvent dans la littérature le problème de paramétrage de MRP est abordé du point de vue d'incertitude de la demande. Cependant, dans le monde industriel, on constate très souvent que les problèmes ne se limitent pas uniquement aux variations liées à la demande, mais aussi aux fluctuations sur les délais. C'est pourquoi dans cette thèse, nous essayons d'apporter des réponses au paramétrage de MRP quand les délais d'approvisionnements sont aléatoires.

La gestion des stocks et la planification des approvisionnements prennent une ampleur très grande avec l'expansion des structures des chaînes logistiques. C'est pourquoi elles sont d'un intérêt évident. Dans cette thèse nous étudions la planification sous incertitudes des délais d'approvisionnement de différentes structures de chaîne logistiques. Nous intéressons aux chaînes d'approvisionnement à structure multi-niveau qui ne sont pas ou peu abordées dans la littérature.

Sous la logique MRP, le temps est divisé en intervalles discrets appelés « *time buckets* ». Afin de se rapprocher de la méthode MRP, dans la présente thèse le délai d'approvisionnement est considéré commettant une variable aléatoire discrète. Dans la même perspective de se rapprocher encore de monde industriel nous supposons que cette variable aléatoire suit une loi de distribution discrète quelconque (connue avant l'optimisation).

Dolgui et Ould Louly ont déjà travaillé sur la paramétrage de MRP pour les systèmes d'assemblage avec des délais d'approvisionnements aléatoires. Dans (Ould Louly et Dolgui 2002), ils considèrent un système d'assemblage avec un produit fini et plusieurs composants. Le but est de rechercher des valeurs optimales de délais planifiés pour un système MRP soumis à des délais incertains, en minimisant les coûts de possession et de retard. Les auteurs partent sur des hypothèses de capacités infinies d'approvisionnements (et donc de délais indépendants de la taille des lots). La demande est supposée constante et celles non satisfaites sont différées. La politique Lot pour Lot est utilisée. Une des difficultés est alors la dépendance entre les stocks de différents composants. Pour le cas où les délais d'approvisionnement suivent la même loi pour tous les types de composants et les coûts

unitaires de stockage des composants sont les mêmes, une généralisation du modèle de Newsboy est proposée. Ce modèle est généralisé à son tour pour le cas de la politique d'approvisionnement à couverture fixe dans (Ould Louly et Dolgui, 2004).

Dolgui et Ould Louly (2002) proposent également un modèle Markovien pour étudier le même problème de planification dynamique multi-période, mais pour le cas plus général où les lois de distribution des délais d'approvisionnements et les coûts unitaires de stockage peuvent être différents. Dans (Ould Louly et Dolgui, 2008) les auteurs développent pour ce modèle une procédure par séparation et évaluation (PSE) pour trouver la solution optimale. Plusieurs bornes inférieures et une borne supérieure de la fonction coût ainsi que des propriétés de dominance permettant réduire l'espace de recherche des solutions ont été développées. Les auteurs présentent des résultats numériques, et montrent l'efficacité de la méthode PSE grâce à la qualité des bornes inférieures, de la borne supérieure et de l'utilisation des propriétés de dominance.

Dans (Ould Louly et *al.*, 2008 a), les auteurs considèrent le même problème que celle de (Ould Louly et Dolgui, 2008) mais en modifiant le critère d'optimisation en minimisant le coût de stockage des composants sous la contrainte de niveau de service. Cette étude complète l'étude (Ould Louly et Dolgui, 2008) lorsque le coût de rupture est difficile à mesurer. Une borne inférieure de la fonction coût ainsi que des propriétés de dominance permettant réduire l'espace de recherche des solutions ont été développées. Une autre étude sur le niveau de service pour les systèmes d'assemblage est faite dans Ould Louly et *al.* (2008 b).

Dans la présente thèse, nous généralisons l'étude de Dolgui et Ould Louly (2002), Ould Louly et Dolgui (2002), Ould Louly et Dolgui, (2008), Ould Louly et *al.* (2008 a) et Ould Louly et *al.* (2008 b) pour le cas de chaîne logistiques linaires et des systèmes d'assemblage multi-niveaux.

## 1.6 Conclusion

Dans ce chapitre nous avons présenté des modèles de gestion des stocks et de planification des chaînes logistiques. Nous avons également présenté en détail le principe de la méthode MRP. En effet, cette méthode est volontairement placée dans un environnement certain. Cependant plusieurs incertitudes existent dans la réalité tel que la demande en produits finis et les délais d'approvisionnements auprès des fournisseurs. Nous avons présenté dans ce chapitre les différents paramètres de décision dans la MRP en cas d'incertitudes et ainsi que le problème de paramétrage de MRP sous incertitudes.

Nous allons dans le chapitre suivant nous attarder aux études consacrées aux incertitudes sur les délais d'approvisionnement qui ne sont pas aussi nombreuses que celles concernant les aléas de la demande. Les aléas de délais d'approvisionnement ont longtemps été négligés par rapport aux incertitudes concernant la demande.

# Chapitre 2 : Etat de l'art

## 2.1 Introduction

La planification des approvisionnements des chaînes logistiques (systèmes de production de biens et de services) est sujette à de nombreux travaux de recherche. Face aux exigences des clients en terme de qualité de produit, de niveau de service offert et de prix, les industriels ont besoin d'améliorer la performance de la gestion des leurs entreprises. Les chercheurs essayent de trouver en situation réelle des formulations intéressantes pour proposer des solutions nouvelles et efficaces qui répondent aux besoins des industriels. Cette thèse est une contribution dans ce domaine de recherche qui est passionnant et extrêmement utile à long terme.

Différentes sources d'aléas existent le long de la chaîne logistique. Ces aléas sont dus à des pannes de machines, aux retards de transport, ou encore aux problèmes de qualité. Pour pallier cela, les entreprises utilisent les stocks de sécurité (souvent d'une manière abusive). Selon Lee et Billington (1993), un grand défi actuel est de trouver le moyen, par la planification efficace d'approvisionnements, de contrôler les stocks et les coûts le long de la chaîne logistique, tout en maximisant le service offert aux clients, en tenant compte des aléas.

Beaucoup de travaux existent sur la gestion des stocks et la planification d'approvisionnement dans les conditions d'une demande aléatoire. Ils ont fait avancer la réflexion générale sur la planification des systèmes soumis aux aléas et les méthodes mathématiques correspondantes. Mula *et al.* (2006) présentent un excellent état de l'art. Au contraire, les problèmes de planification pour les systèmes soumis aux aléas des délais d'approvisionnement ne sont pas suffisamment étudiés.

Pourtant c'est important de tenir compte des aléas des délais approvisionnement comme il montre Silver and Zufferey (2005). Ils ont étudié un problème industriel de la gestion des matières premières de la scierie, où la demande est constante et connue, les réapprovisionnements sont effectués pour

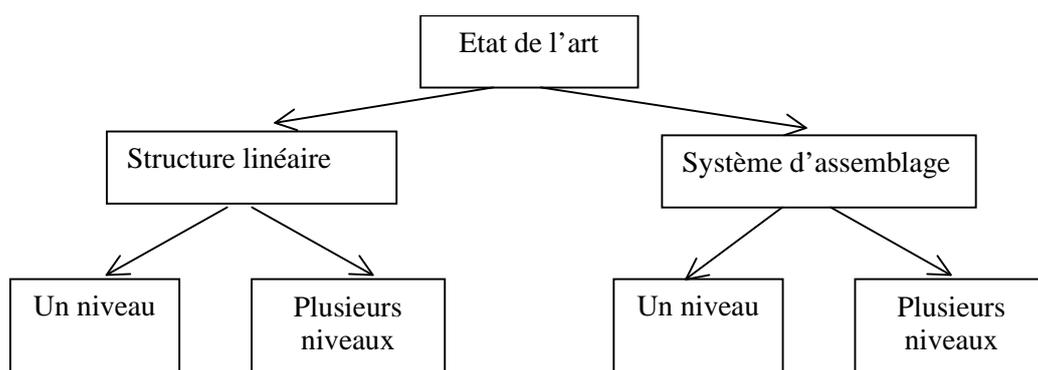
chaque semaine, et les délais d'approvisionnement sont aléatoires.

L'étude des modèles avec des délais d'approvisionnement aléatoires dépend de la structure de la chaîne d'approvisionnement. Nous pouvons identifier trois types de structures de chaîne logistique qui ont été dans la littérature pour ce type de problème: les systèmes simples (un seul fournisseur d'un seul type de composant), les chaînes d'approvisionnement à structure linéaire et les systèmes d'assemblage. Pour ces trois types des chaînes logistiques nous distinguons les structures à un niveau et à plusieurs niveaux.

En effet, pour les cas des structures à plusieurs niveaux il y a une dépendance entre les niveaux : un retard ou une avance de livraison à un niveau donnée créera une perturbation au niveau suivant. Pour le cas des systèmes d'assemblage il y a en plus la dépendance entre les stocks des composants nécessaires pour assembler le même produit, cela vient de fait que l'assemblage ne peut pas être effectué que lorsque tous les composants sont présents.

L'objectif du chapitre est donc de passer en revue les travaux qui s'intéressent à notre problématique afin d'identifier les futures directions de recherche. Dans ce que suit, nous présentons un état de l'art sur les travaux concernant la gestion des stocks avec délais d'approvisionnement aléatoires en classant ces études suivant la structure de la chaîne d'approvements. Pour chaque type d'études nous présentons : le type de méthode de résolution utilisée (approche approximative où exacte), le type de la modélisation de délais d'approvisionnement aléatoires (continu ou discret), la structure de la chaîne d'approvisionnement et les résultats obtenus.

La figure 2.1 présente une classification de l'état de l'art des problèmes étudiés dans ce chapitre sous forme d'un diagramme.



**Figure 2.1** : Classification de l'état de l'art des problèmes étudiés

## 2.2 Chaîne d'approvisionnement à structure linéaire

### 2.2.1 Structure à un niveau

#### 2.2.1.1 Délai aléatoire et demande connue

Kaplan (1970) était parmi les premiers chercheurs qui ont étudié ce type de problème. Il a traité le cas simple à un seul composant et il a modélisé le délai d'approvisionnement comme une variable aléatoire discrète. Il a proposé un modèle de programmation dynamique sur un horizon fini pour trouver la politique de gestion des stocks optimale. L'auteur a montré sous les hypothèses (i) il n'y a pas chevauchement des ordres et (ii) le délai d'approvisionnement ne dépend pas de la taille de la commande que les politiques de gestion des stocks lorsque le délai d'approvisionnement est aléatoire sont assez similaires à celles connues lorsque le délai d'approvisionnement est déterministe.

Whybark et Williams (1976), en utilisant une simulation, ont montré que l'utilisation du délai de sécurité est plus efficace que l'utilisation de stock de sécurité quand il y a des incertitudes sur le temps d'approvisionnement. Une conclusion contraire est donnée par Grasso et Taylor (1984) dont la simulation montre qu'il est plus prudent d'utiliser les stocks de sécurité pour les deux incertitudes temps et quantité. On peut donc conclure que tout dépend des caractéristiques du système étudié (voir plus loin nos commentaires sur la publication de Molinder, 1997).

Weeks (1981) a développé un modèle mathématique pour le cas de système à un seul niveau et un seul type de composant avec le délai d'approvisionnement est aléatoire et la demande est déterministe. Le coût de stockage et le coût de rupture sont pris en compte. L'auteur a montré que le modèle est équivalent au modèle de marchand de journaux « Newsboy model ».

Kalpakam *et al.* (1981) ont traité un problème de planification de la maintenance préventive d'une machine quand la date de panne est aléatoire qui suit une loi de distribution continue et quelconque. Les auteurs cherchent la date d'intervention optimale afin de réduire le coût total. Ce résultat peut être adapté aux problèmes de planification d'approvisionnement.

Kim *et al.* (2004) ont proposé un modèle où la demande est constante et le délai d'approvisionnement suit la distribution d'Erlang pour un système à un seul composant. Une solution approchée est proposée. Par ailleurs, les auteurs ont lancé une intéressante conjecture que le comportement de modèle de gestion des stocks, pour le cas où la demande et les délais d'approvisionnement sont aléatoires, peut être calculé à partir des trois modèles suivants : (i) la demande et le délai sont déterministe, (ii) la demande est aléatoire mais le délai est déterministe, et (iii) le délai est aléatoire mais la demande est déterministe.

He *et al.* (2005) ont étudié l'impact de délai d'approvisionnement lorsque la demande est constante. Dans cette étude le délai d'approvisionnement est aléatoire et borné, et la politique d'approvisionnement est à quantité économique. Les auteurs ont montré que le coût varie linéairement en fonction de la déviation de délai.

### **2.2.1.2 Délai et demande aléatoires**

Dirickx et Koevoets (1977) ont traité la planification d'un système à un niveau et un composant lorsque la demande en produit fini et le délai d'approvisionnement en composants sont aléatoires. La demande suit une loi de Poisson composée et le délai d'approvisionnement suit une loi continue quelconque et connue. Les auteurs ont considéré la planification sur plusieurs périodes toute en supposant qu'il n'y a pas chevauchement des ordres, i.e., l'entreprise relance l'ordre chez le fournisseur que si la commande passée est déjà arrivée. Les auteurs ont présenté analytiquement le coût moyen à minimiser en utilisant le modèle continu et la politique  $(s, S)$ . Ce coût moyen est composé du coût moyen de lancement des ordres, du coût moyen de stockage de composants et du coût moyen de rupture en produit fini. On peut citer deux autres études qui sont très proches de celle-ci : Gross *et al.* (1971) et Gross et Harris (1973). La différence entre ces trois travaux c'est la manière de modélisation de délai d'approvisionnement et la demande. Gross *et al.* (1971) et Gross et Harris (1973) supposent que le délai d'approvisionnement dépend de nombre d'ordres urgents tandis que Dirickx et Koevoets (1977) supposent que le délai d'approvisionnement suit une loi de distribution indépendante de l'ordre et ils considèrent que la demande suit une loi de Poisson composée et non pas une loi de Poisson simple.

Liberatore (1979) a proposé une extension du modèle de gestion à quantité économique « Economic order quantity » (EOQ) lorsque le délai d'approvisionnement est aléatoire. L'auteur a généralisé le coût d'un seul cycle sur un horizon de plusieurs cycles tout en supposant qu'il n'y a pas chevauchement des ordres (crossover). Il a montré l'existence et l'unicité de la solution et que les paramètres optimaux doivent satisfaire un système de deux équations mais sans donner une expression analytique de ces paramètres. Sphicas (1982) a montré que ce modèle donne lieu à deux cas distincts et il a développé des bornes pour les variables de décision pour chaque cas. Georghios *et al.* (1982) ont résolu ce système et ont donné le résultat optimal, mais en supposant que le délai d'approvisionnement varie dans un intervalle fini et pas infini comme il a été supposé par Liberatore (1979).

Mohebbi (2003) a étudié ce problème sous la condition que demande suit une loi de Poisson et le délai d'approvisionnement suit une loi d'Erlang en modélisant cela par une chaîne de Markov à temps continu. Les fournisseurs sont soit disponibles, soit ne le sont pas, à tout point de l'échelle du temps. Il y a donc des interruptions dans les approvisionnements, ce qui provoque des impacts négatifs sur le niveau de service. Les demandes non satisfaites sont perdues. Le système de gestion des stocks considéré est un système à point de commande et quantité fixe de commande  $(s, Q)$ . On ne peut

effectuer qu'une commande à la fois et ainsi des expressions analytiques exactes pour la fonction coût sont obtenues quand la demande est décrite par une variable aléatoire exponentielle.

Robinson *et al.* (2001) ont montré que lorsqu'il y a chevauchement des ordres (crossovers) c'est mieux d'utiliser la loi de distribution de stock que la loi de distribution de la demande durant le délai d'approvisionnement (LTD). Bradley et Robinson (2005) ont confirmé ce résultat lorsque le délai d'approvisionnement est aléatoire en utilisant une approximation des stocks par une loi Normal au lieu de l'approximation du LTD par une loi Normal.

Axsäter (2006) a étudié la gestion des stocks quand le délai d'approvisionnement auprès du fournisseur est aléatoire. L'auteur a montré quelles modifications doivent être faites dans les modèles de gestion des stocks traditionnels qui supposent ce délai constant. Due à la difficulté d'évaluer la variation aléatoire du délai d'approvisionnement par rapport à la variation de la demande, l'auteur a intégré les variations des délais dans les calculs de la demande et il a étudié la demande moyenne au cours de délai aléatoire. Ainsi le problème est devenu équivalent aux problèmes de gestion des stocks quand la demande est aléatoire. Cette idée reste facile à implémenter pour le système simple à un seul type de composants. Lorsqu'il s'agit d'un système d'assemblage où tous les délais d'approvisionnement sont aléatoires, une dépendance est créée entre les stocks de composants. Étant donnée cette dépendance, le problème devient plus complexe.

Namit (2007) a étudié le même problème que Weeks (1981) sur un horizon de planification d'une année et lorsque la demande et le délai sont aléatoires. L'objectif est de déterminer le point de commande  $R$  et le stock de sécurité  $S$ . L'auteur a formulé le coût annuel à minimiser composé de coût de stockage et de coût de lancement d'ordre annuel. L'optimisation est faite sous la contrainte de niveau de service. L'auteur a proposé une approche analytique basée sur la considération de la LTD.

### **2.2.2 Structure à plusieurs niveaux**

Lorsque la nomenclature de produit fini est à plusieurs niveaux, il y a de dépendance entre les stocks de tous les niveaux. Un retard ou une avance à un niveau donné perturbe les niveaux ascendants. Cette dépendance rend la modélisation de ce type de structure de chaînes logistiques plus complexe. Si on applique les modèles connus pour le cas des systèmes à un seul niveau pour ces systèmes, à plusieurs niveaux, niveau par niveau, i.e., on ne tient pas compte de la dépendance entre les niveaux, on risque à créer un sur-stockage dans certains niveaux et une rupture dans d'autres.

Yano (1987 a) a traité analytiquement le problème de planification lorsque les délais d'approvisionnement sont aléatoires et lorsque la structure de la chaîne logistique est à plus d'un niveau. Elle a proposé une approche analytique pour déterminer les délais d'approvisionnement optimaux pour le cas de chaîne d'approvisionnement à structure linéaire à deux niveaux lorsque les délais d'approvisionnement des composants sont aléatoires et la demande en produit fini est connue. Les délais d'approvisionnement sont supposés suivre une loi de distribution continue quelconque mais

connue. Le coût considéré est la somme des coûts de stockage des composants et le coût de rupture de produit fini. Les résultats expérimentaux montrent que les délais de sécurité de niveau 1 et de niveau 2 peuvent être négatifs. Ce résultat vient de la définition du délai de sécurité qui est la différence entre le délai planifié et le délai moyen. L'auteur a considéré le coût de stockage pour chaque composant lorsque ce dernier est présent avant la date prévue. Cependant, si ce composant est en retard l'auteur a ignoré le coût de retard ou le coût de replanification engendré par ce retard.

Un problème similaire a été étudié par Yano (1987 b) mais avec un coût de retard à chaque niveau. Ce coût représente en fait le coût de replanification si le composant est livré après la date prévue de livraison. La fonction objectif est donc la minimisation de la somme de coût moyen de stockage des composants, de coûts de retard à chaque niveau et de coût de rupture en produit fini. L'auteur s'est limité à l'étude des chaînes d'approvisionnement à structure linéaire à deux et à trois niveaux. Cela est dû à la difficulté de modéliser analytiquement le problème si le nombre de niveau dépasse trois.

Pour surmonter la difficulté de trouver une expression analytique de la fonction objectif du problème présenté par Yano (1987 b), Elhafsi (2002) développe une modélisation récursive qui permet l'évaluation de la fonction objectif pour n'importe quelle nombre de niveaux dans la chaîne logistique. L'auteur a montré que la fonction objectif est convexe et donc n'importe quel algorithme d'optimisation classique peut résoudre le problème. Par contre, lorsque le nombre de niveau dans la structure de la chaîne d'approvisionnement est grand les difficultés apparaissent. Ceci est dû à la complexité polynomiale de la représentation récursive de la fonction objectif. L'auteur propose donc une heuristique qui permet de trouver la solution pour le cas des chaînes d'approvisionnement à plusieurs niveaux en divisant le système en un nombre de niveaux. La solution approchée peut donc être obtenue rapidement.

Nous citerons également une étude intéressante de Kim *et al.* (2006) sur l'effet de coup de fouet (bullwhip effect) pour le cas d'une chaîne logistique à structure linéaire avec les délais d'approvisionnement et la demande aléatoires. Les auteurs ont supposé que les délais d'approvisionnement sont *i.i.d* pour tous les niveaux et tous les périodes. L'effet de coup de fouet est un phénomène d'amplification de la variabilité de la demande lorsque l'on remonte vers l'amont de la chaîne logistique. Les auteurs ont utilisé une méthode statistique pour montrer la contribution de variabilité des délais d'approvisionnement sur l'amplification de la variation de la demande pour les niveaux antérieurs et comment cette variabilité peut aggraver encore plus l'effet de coup de fouet. Les auteurs ont aussi étudié l'effet du partage de l'information concernant. Avec le partage de l'information, l'effet de coup de fouet n'augmente que d'une manière linéaire en fonction du nombre des niveaux tandis que dans le cas contraire, d'une manière exponentielle.

### 2.2.3 Perspectives de recherche

Le tableau 2.1 récapitule les modèles utilisés pour la planification de chaînes logistiques à structure linéaire en fonction de (i) problèmes traités et (ii) approches utilisées.

On peut conclure pour cette partie que les études analytiques, qui sont intéressées au problème de planification d'approvisionnement lorsque la chaîne logistique est à structure linéaire et lorsque le délai d'approvisionnement est aléatoire, sont limitées à 2 ou 3 niveaux maximums. Les articles qui ont traité le problème avec plus de niveaux utilisent des heuristiques ou la simulation. Certaines études ont traité analytiquement le problème lorsque la demande et le délai sont aléatoires. Ces travaux ont considéré la demande durant le délai d'approvisionnement (lead time demand - LTD ). Ces travaux sont difficiles à généraliser au cas de systèmes à plusieurs niveaux ou des systèmes d'assemblage. Une autre limitation de ces modèles : l'utilisation des lois particulières pour la demande et les délais d'approvisionnement.

Notons également qu'ils modélisent le délai d'approvisionnement par une variable aléatoire continue afin de l'adapter aux modèles de gestion des stocks continus existants. Cette hypothèse de loi continue est souvent gênant (peut amener une erreur d'approximation importante) dans la planification d'approvisionnement de type MRP, qui utilisent des paramètres entiers (nombre de période).

Notons également qu'à notre avis, il est plus facile de généraliser les méthodes qui considèrent uniquement les délais d'approvisionnement aléatoires au cas quand la demande et les délais sont aléatoires que faire l'inverse. En effet, la planification des chaînes logistiques lorsque les délais d'approvisionnement sont aléatoires dépend de la structure de la chaîne d'approvisionnement. Une fois nous avons un modèle pour une structure d'approvisionnement donnée, il est alors possible de le généraliser à la demande aléatoire tandis que les modèles avec délai fixe et demande aléatoire doivent complètement refaites pour tenir compte de la structure.

Ainsi, dans cette thèse nous focalisons sur la planification optimale pour le cas de chaîne d'approvisionnement à structure linéaire quand le délai d'approvisionnement suit une loi discrète quelconque et dans le cas de demande fixe.

**Tableau 2.1** : Classification des résultats pour le cas de chaînes à structure linéaire

Publications		Approche utilisée	Résultats
Kaplan (1970)	Un niveau, coût de lancement, stockage et rupture	Programmation dynamique	Stock de sécurité
Dirickx et Koevoets (1977) Gross <i>et al.</i> (1971) Gross et Harris (1973) Schellhaas (1974)	Un niveau, coût de lancement des ordres, de stockage et de rupture	Modèle de markov, Formule complexe de la politique (s, S)	Politique optimale de type (s, S)
Liberatore (1979) Sphicas (1982) Georghios <i>et al.</i> (1984)	Un niveau, coût de stockage et pénurie	extension du modèle EOQ	Extension du modèle EOQ
Weeks (1981)	Un niveau, Coût de stockage et de rupture	Programmation non linéaire	Politique optimale de type Marchand de journaux
Whybark and Williams (1976) Grasso and Taylor (1984)	Multi niveaux, coût de stockage et niveau de service	Simulations	Stock et délai de sécurité
Yano (1987 a b)	Deux et trois niveaux, coût de stockage, de replanification, de rupture	Programmation non linéaire	Délai de sécurité
Elhafsi (2002)	Plusieurs niveaux, coût de stockage, de replanification, de rupture	Programmation non linéaire et heuristique	Délai de sécurité
Robinson <i>et al.</i> (2001) Bradley et Robinson (2005)	Un niveau, chevauchement des ordres	Etude statistique et mesures de performance	Le stock de sécurité
Mohebbi (2003)	Un niveau, Coût de lancement, stockage et rupture	Chaîne de Markov, Etude statistique	Expression analytique exacte de la fonction coût
Kim <i>et al.</i> (2004)	Un niveau, Coût de lancement, stockage et rupture	Approche régressive, Borne inférieure et supérieure de politique (s,Q)	Délai de sécurité, solution approchée
He <i>et al.</i> (2005)	Un niveau, Politique EOQ	Etude statistique	Impact de délai aléatoire sur le coût total
Axsäter (2006)	Un niveau, LTD	modèle de gestion de stock lorsque le délai est déterministe	Equivalence avec les modèles avec demande aléatoire
Kim <i>et al.</i> (2006)	Plusieurs niveaux Politique (R, S)	étude statistique, partage de l'information	Effet de coup de fouet
Namit (2007)	LTD, cout de stockage, de lancement et contrainte de niveau de service	étude statistique	Détermination des paramètres de la politique (R,Q)

## 2.3 Systèmes d'assemblage

### 2.3.1 Structure à un niveau

C'est Harrison (1973) qui a été à l'origine de cette étude, l'auteur a introduit le modèle de file d'attente d'un système d'assemblage avec plusieurs composants et un seul serveur (c'est-à-dire, un seul type de produit fini). L'auteur a montré que si les arrivés sont indépendants et la capacité de serveur pour l'assemblage est limitée alors le système est instable. La stabilité du système n'est garantie que lorsque la capacité d'assemblage n'est pas limitée. Il y a également d'autres modèles proches de Harrison (1973) qui analysent les performances des systèmes d'assemblage sans chercher des algorithmes d'optimisations. Som *et al.* (1994) ont étudié un problème similaire mais seulement à deux composants. L'arrivée des composants (délais d'approvisionnement) suit une loi de Poisson et la capacité d'assemblage est supposée infinie. Les auteurs ont montré que le flux de la sortie du serveur suit un processus de renouvellement (Markov renewal process). Dans Wilhem et Som (1998), la demande en produit fini, d'une unité à chaque fois, arrive suivant un processus de Poisson. Les auteurs considèrent le délai d'approvisionnement en produit fini qui est égal aux délais maximums de délais d'approvisionnement de composants et ils montrent que l'évolution du stock du produit fini peut être décrite par un processus de renouvellement. Ils montrent comment dans ce cas la distribution de probabilité du niveau de stock de produit fini peut être obtenue. L'inconvénient de ce model est que les auteurs ne considèrent pas la dépendance entre les stocks des composants et ainsi l'optimisation d'approvisionnement des stocks dépendants des composants est remplacée par l'optimisation d'approvisionnement de produit fini.

Yano (1987 c) considère le système d'assemblage à deux niveaux avec un seul type de composants au niveau 1 et deux types de composants au niveau 2 dans le cas où les délais d'approvisionnement sont aléatoires et continus. Le problème est de trouver les délais d'approvisionnement en minimisant le coût de stockage et le coût de rupture. Un algorithme a été développé en exploitant des propriétés de la fonction objectif. Les résultats numériques montrent que les délais de sécurité peuvent être négatifs quand le coût de stockage unitaire des composants est beaucoup plus élevé que le coût de rupture unitaire. Gurnani *et al.* (1996) étudient le même problème que Yano (1987 c) et il considère la planification sur deux périodes. Dans leur modèle, les composants commandés sont livrés durant la même période avec une probabilité donnée ou une période plus tard. Les auteurs ont donné une règle optimale pour ce type de système. Une généralisation de ce modèle, au cas plus de plus deux composants, ou de plus de deux périodes nous parait difficile, car la fonction objectif de ce modèle a une expression qui croît exponentiellement avec le nombre de réalisation possibles pour le délai d'approvisionnement.

La simulation est la méthode la plus fréquente pour la résolution de problème d'incertitudes des délais. Molinder (1997) propose une procédure de simulation pour trouver un stock de sécurité et un délai de

sécurité efficaces dans un système MRP quand la demande et les délais sont aléatoires. Le résultat de son étude montre qu'en cas de grande variabilité de la demande et faible variabilité des délais, les coûts de stockage et de retard sont minimisés par l'utilisation de stocks de sécurité. Par contre quand les deux variabilités sont grandes, alors il vaut mieux avoir recours à des délais de sécurité. Il semblerait également que la structure des produits n'ait pas d'influence importante sur ces résultats. Gupta et Brennan (1995) ont étudié les systèmes MRP par simulation et ont montré que les aléas des délais d'approvisionnement ont une grande influence sur les coûts. Ho et Irelande (1998) montrent que les incertitudes sur les délais affectent la stabilité d'un système MRP quelle que soit la méthode de lotissement utilisée. Les statistiques provenant des simulations faites par Bragg *et al.* (1999) montrent que les délais d'approvisionnement affectent considérablement les stocks.

Tang et Grubbström (2003) considèrent le même problème que Yano (1987 c). Ils ont utilisé la transformation de Laplace pour trouver les propriétés des délais d'approvisionnements qui permettent de déterminer la solution optimale. L'avantage de cette étude par rapport à celle de Yano (1987 c) est qu'elle est plus facile à l'étendre pour le cas plus général d'un système d'assemblage à plusieurs composants. Avec la même approche, Tang et Grubbström (2005) ont traité le choix de fournisseurs. Et dans (Tang *et al.*, 2007), ils ont étudié le problème de recyclage et de reproduction d'un et de deux types de composants quand le délai de désassemblage et le délai d'approvisionnement sont aléatoires. L'objectif est de déterminer à qu'elle date il faut commencer le désassemblage afin de minimiser le coût du stockage de composants et le coût du rupture. Le modèle de Newsboy a été utilisé pour déterminer la date de désassemblage. Il a noté que le problème à un seul type de composant est équivalent, de point de vue modélisation de la fonction coût, au problème de planification d'approvisionnement d'une chaîne linéaire à deux niveaux et celle de deux types de composants au problème d'assemblage de deux type de composants dont chacun d'entre eux nécessite un seul type de composant pour sa fabrication.

Kumar (1989) présente une étude générique pour le cas de système d'assemblage à un niveau où plusieurs composants sont nécessaires pour assembler un seul type de produit fini. L'auteur considère la planification sur une seule période quand les délais d'approvisionnement sont aléatoires et la date de livraison ainsi que la quantité voulue sont connues. Le problème est de déterminer la date de lancement des ordres en minimisant le coût total composé de coût du stockage et du coût de rupture. L'auteur propose une solution optimale pour des cas particuliers de distributions des délais d'approvisionnement (exponentielle, uniforme, et normale).

Une autre étude intéressant est publiée dans Chu *et al.* (1993), elle concerne la planification des réapprovisionnements sur une période pour les systèmes d'assemblage à un seul niveau quand les délais d'approvisionnement sont aléatoires et suivent des lois continues quelconques mais connues. Le critère considéré est l'espérance mathématique de la somme du coût de rupture en produit fini et du

coût de stockage des composants. Les auteurs montrent la convexité du coût moyen et proposent ainsi un algorithme itératif qui le minimise.

Dolgui *et al.* (1995) proposent un modèle analytique pour la planification des systèmes d'assemblage dans le cas où la demande est constante et les délais d'approvisionnements des composants sont aléatoires et la politique d'approvisionnement considéré est la politique lot pour lot. Il y a plusieurs types de produits et pour assembler chaque produit plusieurs types de composants sont nécessaires. Les auteurs ont pris en considération le coût de stockage et les coûts de rupture. Une approche basée sur l'accouplement d'un modèle de programmation linéaire en nombres entiers avec une simulation et des heuristiques a été développée. Ce modèle calcule le nombre des composants de chaque type à commander au début de chaque période et le nombre de produits de chaque type à assembler. Le même problème a été étudié par Proth *et al.* (1997), mais les composants à commander et les produits à assembler sont choisis sur la base d'une heuristique.

Fujiwara et Sedarage (1997) ont étudié la politique quantité de commande point de commande  $(Q, r)$  pour le cas de système d'assemblage d'un seul type de produit et plusieurs composants où les délais d'approvisionnement sont aléatoires. Les auteurs supposent que la demande est constante et connue et la capacité d'assemblage est infinie. Les auteurs considèrent le coût de stockage de composants, le coût de rupture en produit fini et le coût de setup. Ils ont développé un modèle continu pour cette politique  $(Q, r)$  : quand le stock arrive au niveau  $r_i$ , pour le composant  $i$ , un lot de taille  $Q$  de composant de type  $i$  est lancé. La valeur de  $Q$  est la même pour tous les composants. La décision à prendre donc est de trouver le point de lancement des ordres  $r_i$  pour chaque composant  $i$  et la taille de lot  $Q$ . La fonction objectif obtenue est non linéaire. Les auteurs ont décomposé le problème en sous problèmes et la solution a été obtenue numériquement. Ce modèle est proche du problème de Ould Louly *et al.* (2008 a), mais il y a plusieurs approximations additionnelles : le modèle est continu au lieu de discret (pourtant, le nombre de composants en stock et le nombre de lots fournis sont des variables discrètes dans des systèmes de fabrication manufacturière), la solution est numérique au lieu d'une solution optimale analytique. Fujiwara et Sedarage ont donné les résultats expérimentaux pour seulement 2 composants (le temps de calcul n'était pas fourni), et il est difficile d'interpréter les résultats dans un contexte industriel, par exemple, pour un coût de rupture de 350, la solution optimale est  $r_1 = -1.3$ ,  $r_2 = 2.9$  et  $Q = 90.4$ , pour un coût de rupture de 250, la solution optimale est  $r_1 = -6.3$ ,  $r_2 = -2.1$  et  $Q = 92.9$ . Théoriquement ces résultats sont correctes par contre dans la pratique c'est difficile de les appliquer : qu'est ce que signifie exactement une valeur négative et non entière pour un gestionnaire de stock ? Cela confirme la difficulté pour adapter les modèles standards (développés pour des problèmes à un seul composant) dans le contexte des systèmes d'assemblage, particulièrement en cas de délais d'approvisionnement aléatoires.

Song *et al.* (2000) ont considéré la planification d'un système d'assemblage sur une seule période et à un seul niveau. Le produit fini nécessite plusieurs types de composants, approvisionnés chez un ou

plusieurs fournisseurs. Les auteurs considèrent que la demande en produit fini et les délais d'approvisionnement en composants sont aléatoires. L'objectif est donc de minimiser le coût total composé du coût de stockage des composants avant leur assemblage, du coût de rupture en produit fini, du coût d'obtention. Le coût est une fonction aléatoire continue de quantités à commander et de délais d'approvisionnement planifiés. Les auteurs ont montré que cette fonction est convexe en fixant le vecteur de délais d'approvisionnement mais elle n'est pas en général convexe par rapport aux deux variables. Les auteurs ont proposé des heuristiques simples : (1) utilisant le modèle de marchand de journaux, (2) utilisant les délais moyens, (3) utilisant la demande moyenne, (4) utilisant l'algorithme de descente basée sur (1). Les résultats numériques ont montré que l'ignorance de l'incertitude de délais d'approvisionnement peut coûter cher et des heuristiques simples qui tiennent compte de cette incertitude améliorent assez bien la performance.

Bookbinder et Çakanyildirim (1999) ont étudié le problème de gestion des stocks avec une demande constante et un délai d'approvisionnement qui suit une variable aléatoire continue. Ils analysent la politique "quantité de commande point de commande". Deux modèles analytiques ont été proposés et pour chaque modèle, la convexité du coût moyen a été démontrée, le minimum est trouvé et des exemples numériques sont fournis. Çakanyildirim *et al.* (2000) ont étudié le même problème que Bookbinder et Çakanyildirim (1999) mais lorsque le délai d'approvisionnement dépend de la taille de lot. Les auteurs ont traité les deux cas : quand le délai est une fonction linéaire et quand il est une fonction concave de la taille de lot. Ils ont montré la convexité du coût moyen (coût de stockage, de lancement et de rupture) quand les variables de décision sont  $Q$  et  $R$ . Sous certaines conditions sur une corrélation entre le délai et la taille de lot, les auteurs présentent les solutions optimales analytiquement.

Dolgui et Ould-Louly (2002) ont proposé un modèle Markovien pour un problème de planification dynamique multi périodes. Le but est de rechercher des valeurs optimales de délais planifiés pour un système MRP soumis à des délais incertains, en minimisant les coûts de possession et de retard. Ils partent des hypothèses de capacités infinies d'approvisionnement et de délais indépendants de la taille des commandes. La demande est supposée constante et celles non satisfaites sont différées.

Dans (Ould-Louly et Dolgui, 2002), les auteurs considèrent un système d'assemblage d'un produit fini avec plusieurs types de composants. L'objectif est de minimiser les coûts moyens de possession des composants et de retard de produits finis avec pour variables de décision les délais planifiés des composants. Il est supposé que la demande en produits finis est constante et que les délais de livraison des composants sont aléatoires qui suivent une loi de probabilité discrète quelconque et connue, mais de distribution de probabilité identique pour chacun d'entre eux. De même les différents coûts de stockage sont considérés tous égaux. Le principal paramètre de décision est alors le délai planifié de chaque composant. Le modèle prend en considération les dépendances entre périodes et entre composants. Ils montrent alors que quand le nombre de constituants du produit fini augmente, le délai

planifié augmente aussi. De plus, avec deux composants, la valeur optimale du délai planifié augmente avec le ratio (coût unitaire de retard sur coût unitaire de possession). Quand les critères considérés ci-dessus (le nombre de composants et le ratio) croissent, cela entraîne respectivement des pénuries et des coûts de pénalités de retard. Dans ces deux cas, les solutions optimales requièrent des stocks importants. La solution proposée est une généralisation du modèle de marchand de journaux.

Ould-Louly *et al.* (2008 a) ont généralisé les études de Ould-Louly et Dolgui (2002) pour la planification sur plusieurs périodes des réapprovisionnements des systèmes d'assemblage sans les hypothèses que les distributions de probabilité des délais d'approvisionnement des composants sont identiques et que les coûts de stockage sont égaux. Les auteurs ont proposé une procédure de séparation et évaluation « PSE » pour trouver la solution optimale. Cette procédure est basée sur des bornes inférieures et une borne supérieure de la fonction coût et des propriétés de dominance des variables de décision qui permettent de réduire l'espace de recherche des solutions à chaque étape de la procédure. Les auteurs présentent des résultats numériques intéressants et montrent l'efficacité de la PSE grâce à la qualité des bornes inférieures, de la borne supérieure et des propriétés de dominance.

Ould-Louly *et al.* (2008 b) ont étudié le même problème que Ould-Louly *et al.* (2008 a) mais en remplaçant le coût de rupture du produit fini par un niveau de service. Le coût de rupture est parfois difficile à mesurer et le niveau de service est une caractéristique souvent utilisée dans la gestion des stocks. Les auteurs proposent aussi une PSE pour trouver la solution optimale.

### **2.3.2 Structure multi-niveau**

Les études concernant les systèmes multi-niveau s'intéressent souvent aux problèmes de lotissement. Ho et Lau (1994) ont étudié, par simulation, des différents algorithmes pour faire face aux incertitudes sur les délais d'approvisionnement en regardant l'impact que cela provoquait dans les systèmes MRP. L'essentiel de leur travail a porté sur l'influence des méthodes de lotissement, mais aussi de la structure des produits et de la structure des coûts. Les performances de MRP sont mesurées principalement selon 3 critères : le coût total moyen, un coût de replanification et le nombre total moyen de messages de replanification générés. Parmi cinq méthodes de lotissement, les règles du Silver et Meal (SM), du Part Period Balancing (PPB) et du Wagner et Within (WW) obtiennent des coûts totaux moyens significativement plus faibles que les règles du lot-pour-lot (LFL) et de la quantité économique de commande (EOQ). En ce qui concerne la stabilité, le PPB se révèle être la règle la plus intéressante et celle du LFL la moins performante. De plus, les écarts entre les différentes méthodes se creusent avec l'augmentation de l'incertitude. Il apparaît que, dans le monde industriel où les incertitudes sont inévitables, une règle aussi simple que celle du PPB peut se révéler tout aussi performante qu'une plus complexe comme celle du WW. Cependant, quand la structure du produit ou le délai change, alors les résultats peuvent changer les expériences qui ont été faites avec les règles SM, LFL et EOQ. L'observation faite est que les performances des deux dernières sont alors

inversées, la règle LFL devient plus performante que celle du EOQ. Hegedus et Hopp (2001) ont développé une méthode pratique pour calculer le délai de sécurité pour des composants dans un système d'assemblage avec incertitudes dans le processus d'approvisionnement. Leur approche est basée sur l'utilisation de la méthode MRP avec des capacités de production limitées et une échelle de temps discrète et finie. Ils étudient un système à deux niveaux de production et dont les fournisseurs ont une capacité illimitée. Leurs délais sont des variables aléatoires *i.i.d* et la demande et le taux de production sont déterministes. Le but est de minimiser les coûts de stockage sujets à des contraintes de niveau de service, dans un planning de lancement de production (ici, le niveau de service n'est pas lié au client, mais représente le pourcentage de jobs commençant à temps). Leurs résultats numériques montrent que quel que soit le niveau d'incertitude sur les délais de fournisseurs, un délai de sécurité est utile ne serait-ce que pour garantir une certaine flexibilité. Ils ont aussi regardé l'influence d'un regroupement des lots. Le regroupement a des effets similaires à l'introduction d'un délai de sécurité, mais engendre des coûts de stockage plus importants. Gomaa *et al.* (1999) ont aussi étudié le lotissement pour un système MRP multi-niveau sous délais incertains en proposant un programme mathématique. Leur étude considère trois distributions de délais. Ils ont comparé neuf règles de lotissement et ils en résultent que sous la majorité des conditions, l'algorithme de WW est la meilleure méthode de lotissement. Enfin, Gupta et Brennan (1994) se sont intéressés au sujet dans un environnement MRP avec des structures de produits multi-niveau (21 composants et 6 niveaux) en présence d'incertitudes sur les délais. Dans leurs simulations, ils font varier les délais, la structure des produits (quatre structures) et les paramètres des coûts en étudiant dix méthodes de lotissement. Les résultats montrent que la nomenclature des produits a une influence sur les coûts aussi bien en cas déterministe qu'aléatoire et les incertitudes sur les délais d'approvisionnement des composants se situant au milieu des niveaux de nomenclature du produit affectent plus les performances que celles des autres.

Le seul travail qui s'intéresse à déterminer les délais d'approvisionnements optimaux pour le cas de systèmes d'assemblage à multi-niveau est celui d'Axsäter (2005). L'objectif de ce travail est de trouver les dates de lancement optimales dans le but de réduire le coût total moyen composé d'un coût de stockage des composants et d'un coût moyen de rupture en produit fini. L'auteur ne présente pas la modélisation analytique du problème mais il propose une heuristique pour trouver une solution basée sur la décomposition par niveau. Ce principe de décomposition s'appuie sur la solution optimale pour le système d'assemblage à un niveau. Axsäter a traité deux exemples : le premier exemple est celui d'un système d'assemblage à deux niveaux avec deux types de composant au niveau 1 et trois types de composants au niveau 2 nécessaires pour assembler le premier composant de niveau 1. Le deuxième exemple est un système d'assemblage à trois niveaux avec un seul type de composants au niveau 1, deux types de composants au niveau 2 nécessaires pour l'assemblage du composant de niveau 1 et quatre types de composants au niveau 3 deux pour assembler chaque type de composant au niveau 2.

Les résultats numériques montrent que pour le premier exemple l'erreur maximale est de 1.4%, c'est-à-dire le coût total augmente de 1.4% par rapport à la solution (obtenue par simulation). Pour le deuxième exemple l'erreur peut atteindre 16.6% qui n'est pas négligeable pour un exemple simple de sept types de composants en total. L'auteur a utilisé pour ces exemples des délais d'approvisionnement des composants qui suivent une loi de probabilité continue avec la même distribution de probabilité pour tous les types de composants. De même, les différents coûts de stockage sont considérés tous égaux. L'auteur a mentionné que cette approximation a été utilisée dans le but de simplifier l'obtention de la solution optimale pour des comparaisons. Notons que les mêmes hypothèses ont été faites dans Ould Louly *et al.* (2002), pour un problème identique pour un système d'assemblage à un niveau pour obtenir une généralisation de modèle de Newsboy.

### 2.3.3 Perspectives de recherche

Table 2 récapitule les modèles proposés pour le cas de systèmes d'assemblage en fonction (i) d'approche utilisée et (ii) les résultats.

A notre connaissance, les études qui s'intéressent aux systèmes d'assemblage lorsque les délais d'approvisionnement sont aléatoires utilisent plus souvent une modélisation continue des délais d'approvisionnement. Les seuls travaux qui modélisent les délais d'approvisionnement comme étant des *variables aléatoires discrètes* sont ceux de Dolgui, Ould -Louly, Portmann et Proth. Les études qui se sont intéressées à la détermination des délais d'approvisionnement planifié sont limitées à un seul niveau, le seul travail qui traite le cas de plus d'un niveau c'est celui d'Axsäter (2005) dont les résultats numériques ont montré que l'erreur peut être très grande pour un exemple simple de sept types de composants au total. Ainsi, il est nécessaire de remplir ce manque de recherche et de trouver des solutions optimales ou très proche de l'optimum pour les systèmes d'assemblage multi-niveau et quand le délai d'approvisionnement suit une loi discrète.

**Tableau 2.2** : Classification des résultats pour le cas des systèmes d'assemblage

Publications	Approche utilisée	Résultats
Harrison (1973)	File d'attente	Analyse de performance des solutions
Som <i>et al.</i> (1994)	Modèle markovien de renouvellement	Analyse de performance des solutions
Wilhelm et Som (1998)	Processus markovien	Délai d'approvisionnement de P.F et pas des composants
Yano (1987 c)	Programmation non linéaire	Stock de sécurité
Molinder (1997)	Simulation	Stock et délai de sécurité
Gupta et Bernan (1995) Ho et Ireland (1998) Bragg <i>et al.</i> (1999)	Simulation	Impact des aléas de délais sur le coût et sur la stabilité de MRP
Tang et Grubbström (2003) Tang et Grubbström (2005) Tang <i>et al.</i> (2007)	Transformé de Laplace	Choix des fournisseurs Stock de sécurité
Kumar (1989)	Programmation stochastique	Les dates de lancement des ordres des cas particuliers de distribution (exponentielle, uniforme et normale)
Chu <i>et al.</i> (1993)	Optimum global, algorithme itératif de type gradient	Délais d'approvisionnement planifiés
Dolgui <i>et al.</i> (1995)	Couplage d'un modèle de programmation linéaire en nombres entiers avec une simulation et des heuristiques	Délais d'approvisionnement planifiés
Proth <i>et al.</i> (1997)	Heuristiques et simulation	Détermination des composants à commander et des produits à assembler
Fujiwara et Sedarage (1997)	Politique $(Q, r)$ , programmation non linéaire	Paramètres optimaux $(Q, r)$
Song <i>et al.</i> (2000)	Modélisation analytique et heuristiques	Détermination des quantités à commander et des délais d'approvisionnement
Bookbinder et Çakanyildirim (1999) Çakanyildirim <i>et al.</i> (2000)	Politique $(Q, r)$	Paramètres optimaux $(Q, r)$
Song <i>et al.</i> (2001)	Heuristiques, simulation et procédures récursives	Paramètres optimaux $(Q, r)$
Ould-Louly et Dolgui (2002) Ould-Louly et Dolgui (2004) Ould-Louly <i>et al.</i> (2008 a b)	Modèle de Newsboy, modèle de Markov, PSE	Stock initial optimal pour le cas de modèle multi-période
Ho et Lau (1994) Hegedus et Hopp (2001) Gomaa <i>et al.</i> (1999) Brennan (1994)	simulation	Comparaison des méthodes de lotissement de point de vue stabilité, minimisation de coût, flexibilité
Axsäter (2005)	Technique de décomposition, simulation	Date approchée de lancement d'ordre

## 2.4 Choix de fournisseurs

D'autres problèmes liés aux délais d'approvisionnement aléatoires concernent le choix des fournisseurs.

Les travaux analysant les stratégies multi-fournisseur en présence des délais d'approvisionnement aléatoires se focalisent souvent sur le fractionnement de la quantité des composants à commander entre plusieurs fournisseurs. Supposant que la quantité de commande  $Q$  se fractionne entre  $n$  fournisseurs ayant des délais d'approvisionnement aléatoires, chaque commande  $Q/n$  passée à chacun des  $n$  fournisseurs. L'avantage de placer des commandes auprès des plusieurs fournisseurs lorsque les délais sont aléatoires, est souvent lié avec la réduction de l'écart type pour le temps d'intrè-arrivées de commandes. Sculli et Wu (1981) analysent le cas de deux fournisseurs où les délais de livraison des fournisseurs suivent la même loi de distribution (loi normale). Les auteurs utilisent une intégration numérique pour présenter la valeur moyenne et la variance des délais moyens effectifs des arrivés de deux commandes. L'expression analytique a été présentée par Fong (1992). Sculli et Shun (1990) ont généralisé leur étude pour le cas de  $n$  fournisseurs. Les auteurs présentent les expressions de la valeur moyenne et la variance des délais effectifs. Pan *et al.* (1991), traitent le cas général de  $n$  fournisseurs lorsque les délais d'approvisionnement sont identiquement distribués. Les auteurs ont obtenu une expression analytique lorsque la loi de distribution est uniforme, exponentielle et normale.

Guo et Ganeshan (1995) étudient le problème à  $n$  fournisseurs quand les délais suivent la même loi de distribution (uniforme et exponentielle) avec des caractéristiques identiques. Les auteurs donnent à travers les caractéristiques des délais effectifs le nombre des fournisseurs à choisir. Fong *et al.* (2000) pour un problème similaire mais lorsque la demande suit une loi normale offrent les expressions analytiques de diverses mesures statistiques.

Ramasesh *et al.* (1991) ont étudié le choix entre deux fournisseurs dans le cas de la politique d'approvisionnement  $(R, Q)$  quand la demande est constante et les délais d'approvisionnement sont aléatoires. Les auteurs ont considéré qu'un seul type de matière première et uniquement le cas où les délais d'approvisionnement suivent une loi uniforme ou une loi exponentielle. Par approximation de la fonction coût total (non linéaire) composé de coût de stockage, coût de rupture, coût de setup et coût d'achat, les auteurs déterminent le point de commande  $R$  et la quantité de commande  $Q$  par des calculs numériques. Ramasesh *et al.* (1993) ont généralisé cette étude pour le cas où les délais d'approvisionnements ne sont plus identiques. Chiang et Benton (1994) s'intéressaient au même type de problème lorsque la demande suit une loi normale et les délais d'approvisionnement suivent des lois exponentielles. Le coût de rupture a été supposé indépendant de la durée de rupture et donc il a été remplacé par une contrainte de niveau de service. Mohebbi et Posner (1998) analysent un problème similaire à Ramasesh *et al.* (1991) avec des délais de livraison suivant des lois exponentielles non identiques et des demandes arrivant selon un processus de Poisson. Les analyses numériques ont

montré que la stratégie de partage de la commande entre deux fournisseurs diminue le coût total par rapport à celle à un seul fournisseur (sauf si l'un de deux fournisseur est beaucoup moins performant). Sedarage *et al.* (1999) ont étudié le cas de plusieurs fournisseurs quand la demande et les délais d'approvisionnement sont aléatoires (non identiques). Les auteurs s'intéressaient à la politique d'approvisionnement  $(s, Q)$  et ont déterminé à travers des calculs numériques le nombre optimale de fournisseurs minimisant le coût total.

Arda et Hennet (2006) ont utilisé la théorie de file d'attente pour un modèle théorique pour un problème similaire avec un détaillant et plusieurs fournisseurs. Il y a un seul type de produit fini. Les délais d'approvisionnement sont aléatoires et suivent une loi exponentielle et la demande en produit fini suit une loi de Poisson. Le client lance une commande et la demande est satisfaite directement du stock de détaillant sinon elle est retardée. La politique d'approvisionnement  $(S-1, S)$ . L'objectif de détaillant est de minimiser les coûts moyens de stockage et de rupture. Les auteurs proposent de construire une politique d'approvisionnement, définie par une position optimale de stock de référence et une règle pour choisir le fournisseur pour chaque ordre. Les résultats numériques montrent que l'avantage économique pour le détaillant est de lancer les ordres à plusieurs fournisseurs plutôt qu'un seul.

Pour plus de discussions, Minner (2003) fournit une revue de littérature sur cette problématique de choix de fournisseur.

## **2.5 Chevauchement des ordres (« crossover »)**

Généralement, une formulation analytique de modèle de gestion des stocks sur plusieurs périodes sous l'incertitude de délai d'approvisionnement est très complexe. A part la difficulté lié à la structure de chaîne d'approvisionnement, il y a aussi le problème dû au chevauchement des ordres connue dans la littérature anglo-saxon sous le nom « crossover » : les commandes peuvent ne pas être reçus dans le même ordre dans lequel elles sont placées. Le problème de chevauchement des ordres a été examiné dans (Hadley et Whitin, 1963).

La possibilité de chevauchement des ordres à la suite de l'indépendance des délais a reçu peu d'attention dans la littérature. Zalkind (1978) a donné un aperçu sur les conséquences de chevauchement des ordres sur la politique de gestion des stocks.

Des conséquences pratiques du chevauchement des ordres sont mises en évidence dans Riezebos (2006). L'auteur donne une explication détaillée de ce phénomène. Il a montré qu'une condition nécessaire pour le phénomène de chevauchement existe est la présence des incertitudes des délais d'approvisionnement. L'auteur classe l'incertitude de délais d'approvisionnement en deux catégories : dynamiques et statiques. L'incertitude dynamique contient une composante systématique, par exemple, en raison des influences saisonnières des différents fournisseurs pour les commandes

successives. Les incertitudes statiques ne contiennent pas une telle composante systématique. Une bonne discussion sur les modèles utilisés dans le cas de délai d'approvisionnement aléatoire est présentée dans (Zipkin, 2000). L'auteur traite le cas où la demande et les délais sont aléatoires pour le cas d'un système simple à un seul fournisseur et un seul composant. L'auteur utilise des approximations afin d'adapter les modèles de gestion des stocks quand la demande et les délais sont aléatoires aux modèles quand la demande est aléatoire et le délai est fixe et constant. Zipkin (2000) donne une autre classification des délais d'approvisionnement aléatoires. Dans ses travaux, le processus d'approvisionnement est supposé soit exogène et séquentiel (*exogenous sequential supply*) soit exogène et parallèle (*exogenous parallel supply*). Le délai d'approvisionnement est dit exogène lorsqu'il ne dépend pas de l'ordre et il est endogène dans le cas où il dépend des ordres urgents. Le processus est *séquentiel*, lorsque les ordres sont exécutés dans la séquence même où ils ont été générés, ce qui crée une dépendance entre les délais d'approvisionnement successifs. Dans le cas d'un processus *parallèle*, l'ordre peut être servi par un fournisseur parmi plusieurs fournisseurs (serveurs). Le système d'approvisionnement est souvent modélisé comme un système de file d'attente avec un nombre infini de serveurs. Dans ce cas, les délais d'approvisionnement successifs sont indépendants. Un système d'approvisionnement endogène confronté aux aléas est souvent modélisé comme un système de file d'attente avec un nombre fini de serveurs.

Les articles qui s'intéressent à l'incertitude de délai d'approvisionnement souvent ne prennent pas en compte le phénomène de chevauchement des ordres. Deux types de travaux existent concernant le chevauchement des ordres : le premier suppose que les chevauchements sont suffisamment petits et ils sont ignorés. Dans la deuxième, une structure particulière des modèles est proposée permettant d'ignorer les chevauchements des ordres. Les articles qui mentionnent explicitement le premier cas sont, par exemple, Bagchi *et al.* (1986), Eppen et Martin (1988), Tijms Groenevelt (1984) et Zheng (1992). Exemples du deuxième cas sont les modèles de Kaplan (1970), Nahmias (1979) et Ehrhardt (1984) et Zipkin (1986).

Pour simplifier, certains articles supposent que ce phénomène de chevauchement ne peut pas avoir lieu. Dans ce cas, une nouvelle difficulté apparaît: les délais d'approvisionnement successifs deviennent des variables aléatoires dépendant (Lee et Nahmias, 1993).

D'autres travaux établissent des hypothèses qui interdisent l'impact négatif de chevauchement des ordres sur leurs modèles de gestion des stocks. Par exemple, les modèles de Liberatore (1979) considèrent la situation où les commandes ne sont pas interchangeableables. Si une commande est en retard et une autre commande, lancé après, arrive, Liberatore ne permet pas l'utilisation de cette commande déjà arrivée pour satisfaire la demande en attente. Il estime que les ordres ne sont pas anonymes, et il met cette hypothèse afin de justifier l'interdiction de chevauchement des ordres.

Ramasesh *et al.*(1991) tiennent compte dans leur modèle de la possibilité de chevauchement des ordres

lorsque le délai d'approvisionnement suit une loi de distribution exponentielle, la demande et l'intervalle entre les ordres sont constants. Haya *et al.* (1995) étendent ces résultats au cas de demande aléatoire.

Robinson *et al.* (2001) montrent que si la décision sur la taille de lot est basée sur la distribution de la demande durant le délai d'approvisionnement, une probabilité faible de chevauchement des ordres peut créer un coût de stockage très élevé. Dans Bookbinder et Cakanyildirim (1999) et Duran *et al.* (2004), la possibilité de chevauchement des ordres n'est pas considérée. Les auteurs supposent que la différence entre les délais d'approvisionnement des ordres ne dépasse jamais le délai entre les ordres. Cette hypothèse a été utilisée également par Riezebos (2006) (corrolaire 2).

Dans Silver and Zufferey (2005), la demande est constante et connue, les réapprovisionnements sont effectués pour chaque semaine, et les délais d'approvisionnement sont aléatoires. Les délais de différents ordres sont supposés indépendants. Par conséquent, le chevauchement des ordres peut se produire (pas plus de trois dans le modèle proposé). La distribution de probabilité du délai d'approvisionnement change de manière saisonnière. La rupture de matières premières engendre une perte de ventes. Le coût total moyen est composé des coûts moyens de lancement de l'ordre, d'exécution et de coût de rupture. L'objectif consiste à déterminer la date de lancement de l'ordre et la quantité de chaque commande. Les auteurs offrent une expression analytique des coûts pour un cycle en utilisant un modèle discret et emploient une simulation pour estimer le coût total annuel. Pour minimiser ce coût annuel, les auteurs ont développé deux approches heuristiques: un algorithme de descente et une recherche Tabou.

Enfin, Diks et Vanderheijden (1997) ont étudié les algorithmes de génération des délais d'approvisionnement par simulation. Ils ont supposé que le chevauchement des ordres ne peut pas se produire, et par conséquent les délais sont dépendants. Les auteurs ont proposé un processus qui génère les délais d'approvisionnement qui garantissent l'absence des chevauchements des ordres.

## **2.6 Conclusions**

Dans ce chapitre, nous avons essayé de faire un état de l'art concernant la gestion des stocks des systèmes soumis à des aléas de délais d'approvisionnements. L'utilisation des délais de sécurité est très répandue dans ce cas pour pallier aux risques de rupture de stocks. Cependant, c'est une méthode qui peut parfois s'avérer coûteuse. La recherche de solutions efficaces, permettant de limiter les coûts tout en satisfaisant les besoins clients, est donc essentielle.

Avec l'expansion du concept de la chaîne logistique, la fonction de planification des approvisionnements prend une ampleur encore plus importante. C'est pourquoi les problèmes concernant cette gestion sont d'un intérêt évident. Les articles ont été classés suivant la structure de la chaîne logistique étudiée vu que la structure influence beaucoup la complexité du problème. Nous

constatons que la plupart des articles traitent le cas simple à un niveau et un seul type de composant. Les auteurs considèrent souvent la demande aléatoire et ils étudient la demande durant le délai d'approvisionnement afin d'appliquer les méthodes de gestion des stocks connues quand la demande est aléatoire et le délai est connu et fixe.

Pour les problèmes à plusieurs niveaux, l'analyse analytique est presque absente. Vu la complexité du problème, souvent les auteurs font recours aux heuristiques ou à des modèles basés sur la simulation. Dans les études qui ont traité le problème analytiquement, la résolution exacte a été limitée à 2 et 3 niveaux avec un nombre de composant limité. Les méthodes de résolution sont souvent numériques.

Pour le cas de système d'assemblage, c'est encore plus complexe, les études les plus connues s'intéressent aux systèmes à un seul niveau. Pour le cas des systèmes d'assemblage multi-niveau, à notre connaissance, il n'y a qu'un seul article, celui d'Axsäter (2005) qui propose des techniques approchées basées sur une décomposition niveau par niveau. Cette méthode même pour le cas simple avec un nombre très limité de composants et de niveaux n'était pas très performante.

Nous constatons également que les délais d'approvisionnement sont le plus souvent modélisés comme des variables aléatoires continues. Cette modélisation nécessite une discrétisation afin de se rapprocher des méthodes industrielles de planification de type MRP qui considère un environnement temporel discret (le temps est calculé en périodes). Ainsi, c'est mieux de modéliser les délais d'approvisionnement en composants comme des variables aléatoires discrètes qui suivent une loi quelconque.

Dans notre thèse, nous allons généraliser les études qui existent pour les cas des chaînes logistiques à nomenclatures complexes tel que les chaînes à structures linéaires multi-niveau qui fera l'objet du chapitre 3 et les systèmes d'assemblage à deux niveaux qui fera l'objet des chapitres 4 et 5.

Dans notre étude, les délais d'approvisionnement sont discrets et non pas continus, nous modélisons le délai d'approvisionnement par une variable aléatoire discrète qui suit une loi de distribution quelconque (connue d'avance).

Pour les chaînes logistiques complexes tel que les chaînes à nomenclature linéaire multi-niveau, la difficulté de calculs des délais de sécurité réside dans l'interdépendance entre les niveaux. En effet, un retard ou avance d'une commande d'un niveau donné perturbe les niveaux ascendants. Pour le cas de systèmes d'assemblage, plusieurs types de composants sont nécessaires pour produire un produit fini ou semi-fini; et donc, les stocks de différents composants sont dépendants. Quand il y a un retard et rupture d'un composant, automatiquement, il n'y a pas de possibilité d'assembler le produit fini. Quand le processus d'assemblage est arrêté, il y aura stock des autres composants livrés et pas utilisés. Pour les chaînes logistiques tel que les systèmes d'assemblage multi-niveaux, la complexité est liée à la fois à la dépendance entre les stocks des composants et à la dépendance entre les niveaux.

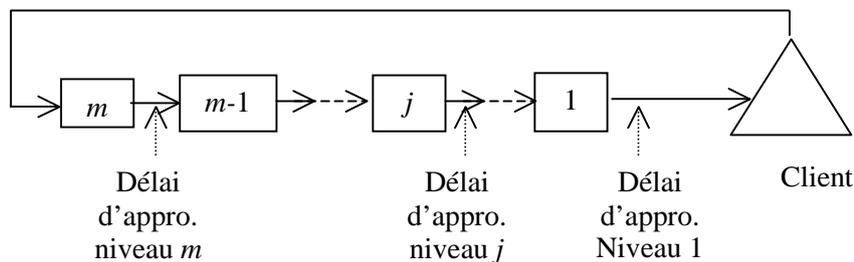
Deux critères sont utilisées dans notre étude, le premier critère est la minimisation du coût total moyen composé du coût de stockage en composants et du coût de rupture en produit fini. En pratique, il est souvent difficile de déterminer le coût unitaire de rupture en produit fini. Ce coût de rupture inclut la pénalité de rupture, mais aussi ce coût imprécis peut endommager l'image commerciale de l'usine ou l'entreprise, perdre des clients, etc. Pour les cas quand les calculs du coût de rupture est impossible, dans nos modèles ce coût est remplacé par une contrainte sur le niveau de service : le deuxième critère est donc la minimisation du coût de stockage sous la contrainte d'un niveau de service.

# Chapitre 3 : Chaîne d'approvisionnements linéaire

## 3.1 Introduction

Nous étudions dans ce chapitre la planification des réapprovisionnements dans une chaîne logistique linéaire à  $m$  niveaux dans le cas où les délais d'approvisionnement à tous les niveaux sont aléatoires et la demande en produit fini et sa date de livraison sont connus (voir la figure 3.1). Les délais d'approvisionnement sont supposés indépendants, suivant des lois discrètes quelconques mais à valeurs bornées.

L'objectif de cette étude est donc de minimiser le coût total moyen dû à l'incertitude des délais d'approvisionnement. Cette incertitude peut engendrer des coûts de stockage des produits ainsi qu'un coût de rupture en produit fini. L'objectif revient donc à trouver les délais d'approvisionnement optimaux permettant de réduire ces coûts.



**Figure 3.1 :** Chaîne d'approvisionnement linéaire à  $m$  niveaux

Nous distinguons deux cas d'étude : (i) le cas de fabrication au plus tôt : à chaque niveau un produit est traité dès qu'il arrive, (ii) le cas avec dates de fabrication : chaque produit est stocké en attendant la date de début de fabrication prévue d'avance pour le niveau correspondant, un coût de retard sera compté si le produit est livré par le niveau précédent après cette date.

Dans le cas (i) de fabrication au plus tôt, nous ne nous tenons compte du coût de stockage qu'au niveau 1 (la dernière phase de fabrication) et nous étudions deux modèles :

Le premier a comme objectif de minimiser la somme du coût moyen de stockage au niveau 1 et du coût moyen de rupture en produit fini. Le deuxième minimise le coût moyen de stockage au niveau 1 tout seul sans tenir compte du coût de rupture. Dans ce deuxième cas pour limiter les ruptures, nous

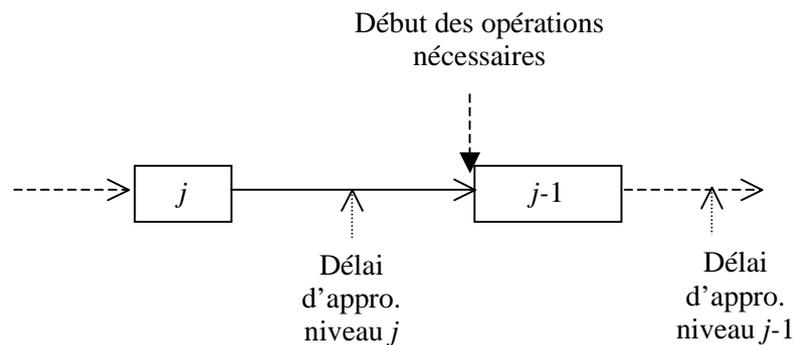
ajoutons une contrainte sur le niveau de service. C'est à dire que le coût de rupture est remplacé par une contrainte sur le niveau de service qui est plus facile à calculer dans les cas pratiques.

Dans le cas (ii) avec dates de fabrication, nous avons également deux modèles :

Le premier vise la minimisation de la somme des coûts moyens de stockage et des coûts moyens de retard de produits de tous les niveaux et le coût de rupture en produit fini. Le deuxième critère minimise la somme des coûts moyens de stockage de tous les niveaux, tout en respectant les contraintes sur le niveau de service à chaque niveau et pour le produit fini.

### 3.2 Le cas de fabrication au plus tôt

Nous supposons ici que pour assurer la demande en produit fini, l'entreprise finale lance la commande chez le fournisseur des matières premières (niveau  $m$ ), le produit commandé traverse toutes les maillons de la chaîne d'approvisionnement pour subir différentes opérations (de niveau  $m$  jusqu'au niveau 1). Lorsque le produit est livré de maillon  $j$  au maillon  $j-1$  ( $j=m, m-1, \dots, 2$ ) il subit toutes les opérations du niveau  $j-1$  dès qu'il arrive, et ainsi de suite (voir la figure 3.2).



**Figure 3.2 :** Cas de fabrication au plus tôt

Nous cherchons soit à minimiser le coût moyen de stockage de produit au niveau 1 plus le coût moyen de rupture en produit fini (problème P1), soit à minimiser le coût moyen de stockage de produit de niveau 1 tout en satisfaisant un niveau de service de client final (problème P2).

Nous présentons la modélisation analytique de chaque problème, les méthodes d'optimisation et des exemples numériques.

#### 3.2.1 Problème P1

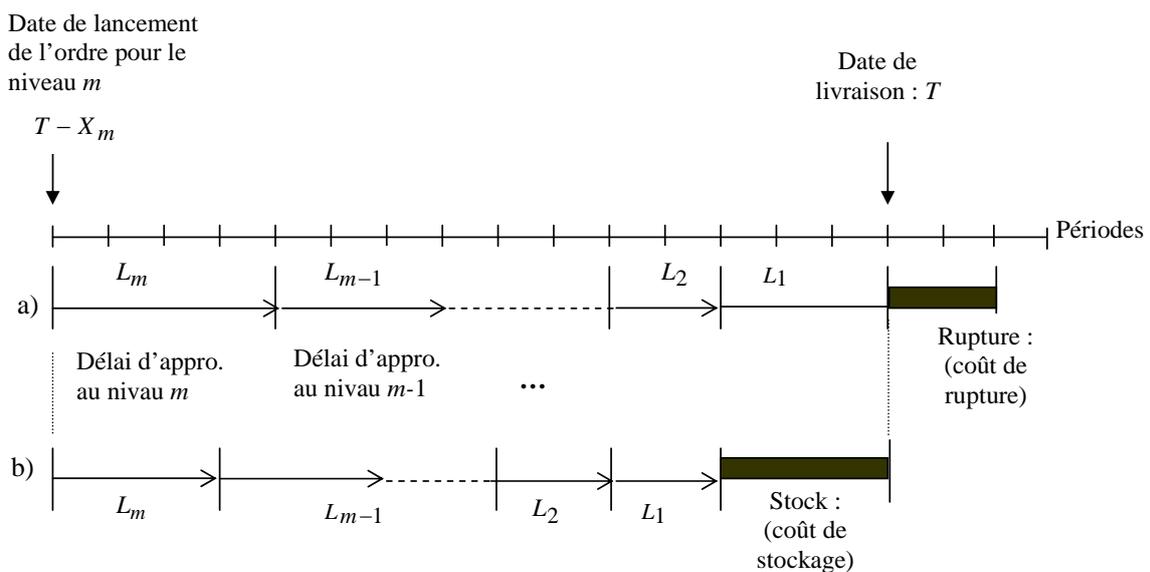
Le problème est donc de trouver la planification optimale minimisant le coût engendré par le stockage au niveau 1 et par la rupture en produit fini. Dans ce qui suit, nous présentons analytiquement le coût total moyen à minimiser.

### 3.2.1.1 Description du problème

Nous introduisons les notations suivantes :

- $c_j$  produits au niveau  $j$ , où  $j = 1, 2, \dots, m$
- $b$  coût de rupture unitaire du produit fini par période
- $h$  coût de stockage unitaire au niveau 1 par période
- $D$  demande en produit fini par période (fixe et connue)
- $T$  date de livraison souhaitée par le client final (connue)
- $L_j$  délai d'approvisionnement du produit  $j$  (variable aléatoire discrète)
- $u_j$  limite supérieure de  $L_j$
- $U = \sum_{j=1}^m u_j$
- $x_j$  délai d'approvisionnement planifié au niveau  $j$
- $F_j(k) = \Pr(L_j \leq k)$
- $(x)^+ : \text{Max}(0, x)$

Comme ici, nous considérons le cas de fabrication au plus tôt : un produit est traité dès qu'il arrive à un nouveau niveau de la chaîne, le problème est de trouver la date de lancement optimale au niveau  $m$  afin de satisfaire la demande  $D$  à la date souhaitée  $T$ , en minimisant le coût total moyen. Si l'ordre de fabrication est lancé trop tôt, le produit final arrive avant la date prévue de livraison au client, et donc le stock au niveau 1 augmente, ce qui induit un coût de stockage complémentaire. Mais, si l'ordre est lancé trop tard, il y aura retard en produit fini, et donc il y aura un coût de rupture en produit fini, proportionnel à la durée de la rupture (voir la figure 3.3).



**Figure 3.3 :** Description du problème P1

### 3.2.1.2 Minimisation des coûts

Nous présentons une modélisation analytique de ce problème. Le critère considéré est la somme du coût moyen de stockage et du coût moyen de rupture en produit fini. Nous montrons que le modèle est équivalent au modèle du marchand de journaux.

**Proposition 3.1** *Le coût total du problème P1 est exprimé comme suit :*

$$C(X_m, L) = Dh \left( X_m - \sum_{j=1}^m (L_j) \right)^+ + Db \left( \sum_{j=1}^m (L_j) - X_m \right)^+ \quad (3.1)$$

où,

$$L = (L_1, \dots, L_j, \dots, L_m)$$

#### Démonstration

Le coût total est égal à la somme du coût de rupture et du coût de stockage du produit fini. Nous allons d'abord exprimer le coût de rupture en produit fini et ensuite le coût de stockage de produit fini.

Pour assurer la livraison du produit fini à la date  $T$  souhaitée, l'entreprise lance un ordre de fabrication pour le niveau  $m$  à la date  $T - X_m$ . Le produit arrivera alors au niveau  $m-1$  à la date  $T - X_m + L_m$ , subira les opérations nécessaires et sera transmis au niveau  $m-2, \dots$ , et ainsi de suite,

jusqu'à ce que le produit final soit prêt à être livré au client à la date :  $T - X_m + \sum_{j=1}^m L_j$ . Il y a donc

stockage du produit fini s'il est présent avant la date  $T$ , i.e.  $\left( T - X_m + \sum_{j=1}^m (L_j) \right) \leq T$ .

C'est-à-dire si  $\left( X_m - \sum_{j=1}^m (L_j) \right) \geq 0$ .

Le coût de stockage associé vaut donc:

$$Dh \left( X_m - \sum_{j=1}^m (L_j) \right)^+ \quad (3.2)$$

De même, il y a rupture en produit fini si :  $\left( X_m - \sum_{j=1}^m (L_j) \right) \leq 0$ .

Le coût de rupture en produit fini est donc égal à :

$$Db \left( \sum_{j=1}^m (L_j) - X_m \right)^+ \quad (3.3)$$

Le coût total est donc égal à :

$$C(X_m, L) = Dh \left( X_m - \sum_{j=1}^m (L_j) \right)^+ + Db \left( \sum_{j=1}^m (L_j) - X_m \right)^+$$

C.Q.F.D.

Ce coût total est une variable aléatoire qui est fonction des variables aléatoires  $L_j$  pour  $j=1, \dots, m$ .

Nous allons nous intéresser au coût moyen  $C(X_m) = E(C(X_m, L))$ .

Afin de simplifier l'écriture des équations sans perdre la généralité de l'approche, nous posons  $D=1$ .

$$\text{Soit } SL = \sum_{j=1}^m (L_j).$$

**Proposition 3.2** *Le coût total moyen du problème P1 est exprimé comme suit :*

$$C(X_m) = h \sum_{k \geq 0} F_{SL}(X_m - k - 1) + b \sum_{k \geq 0} 1 - F_{SL}(X_m + k) \quad (3.4)$$

où,

$$F_{SL}(t) = \Pr(SL = \sum_{j=1}^m (L_j) \leq t) = \sum_{k=m}^t \prod_{\substack{j=1 \\ k_1 + \dots + k_m = k}}^m \Pr(L_j = k_j)$$

### Démonstration

Sachant que :

$$C(X_m) = h_1 E \left( \left( X_m - \sum_{j=1}^m L_j \right)^+ \right) + b E \left( \left( \sum_{j=1}^m (L_j) - X_m \right)^+ \right)$$

Nous calculons tout d'abord :  $E \left( \left( X_m - \sum_{j=1}^m L_j \right)^+ \right)$

$$E(X_m - SL)^+ = \sum_{k \geq 0} k \times \Pr(X_m - SL = k) = \sum_{i \geq 0} \sum_{k=0}^{i-1} \Pr(X_m - SL = i) = \sum_{k \geq 0} \sum_{i > k} \Pr(X_m - SL = i)$$

$$= \sum_{k \geq 0} \Pr(X_m - SL > k) = \sum_{k \geq 0} \Pr(SL \leq X_m - k - 1) = \sum_{k \geq 0} F_{SL}(X_m - k - 1)$$

Ensuite, pour calculer  $E(X_m - SL)^+$  en utilisant la définition suivante :

$$F_{SL}(t) = \Pr(SL = \sum_{j=1}^m (L_j) \leq t) = \sum_{\substack{k=m \\ k_1+\dots+k_m=k}}^t \prod_{j=1}^m \Pr(L_j = k_j)$$

Nous obtenons :

$$E(SL - X_m)^+ = \sum_{k \geq 0} \Pr(SL - X_m > k) = \sum_{k \geq 0} 1 - \Pr(SL \leq X_m + k) = \sum_{k \geq 0} 1 - F_{SL}(X_m + k)$$

C.Q.F.D.

### 3.2.1.3 Optimisation

**Proposition 3.3** La solution optimale de P1 est donnée par les inégalités suivantes:

$$F_{SL}(X_m^* - 1) \leq \frac{b}{b+h} \leq F_{SL}(X_m^*) \quad (3.5)$$

#### Démonstration

Soit la fonction  $G(X_m)$  défini comme suit :

$$G(X_m) = C(X_m + 1) - C(X_m)$$

Nous obtenons donc les deux relations suivantes :

$$\begin{cases} G(X_m) = hF_{SL}(X_m) + b(F_{SL}(X_m) - 1) \\ G(X_m - 1) = hF_{SL}(X_m - 1) + b(F_{SL}(X_m - 1) - 1) \end{cases}$$

La valeur de la solution optimale  $X_m^*$  doit satisfaire les deux conditions suivantes :

$$\begin{cases} G(X_m) = hF_{SL}(X_m) + b(F_{SL}(X_m) - 1) \geq 0 \\ G(X_m - 1) = hF_{SL}(X_m - 1) + b(F_{SL}(X_m - 1) - 1) \leq 0 \end{cases}$$

Donc, la solution optimale doit satisfaire les inégalités suivantes :

$$F_{SL}(X_m^* - 1) \leq \frac{b}{b+h_1} F_{SL}(X_m^*)$$

C.Q.F.D.

Notons, que nous sommes arrivés au modèle discret du marchand de journaux (Porteus, 1990; Ould Louly et Dolgui, 2002).

**Corollaire 3.1**

*La variable  $X_m$  qui donne le coût moyen minimal doit vérifier l'inégalité suivante :*

$$m \leq X_m \leq U \tag{3.6}$$

**Démonstration**

Pour démontrer ce résultat simple il suffit de remarquer que :

si  $X_m > U$  , alors :

$$F_{SL}(X_m \geq U + 1) = 1$$

Donc :

$$F_{SL}(X_m - 1) = 1 \tag{3.7}$$

La relation (3.7) n'est vérifiée que lorsque  $h=0$ . Si le coût de stockage est nul alors toute solution  $X_m \geq U$  est optimale et donne le même coût.

Si  $X_m < m$  , alors :

$$F_{SL}(X_m \leq m - 1) = 0 \tag{3.8}$$

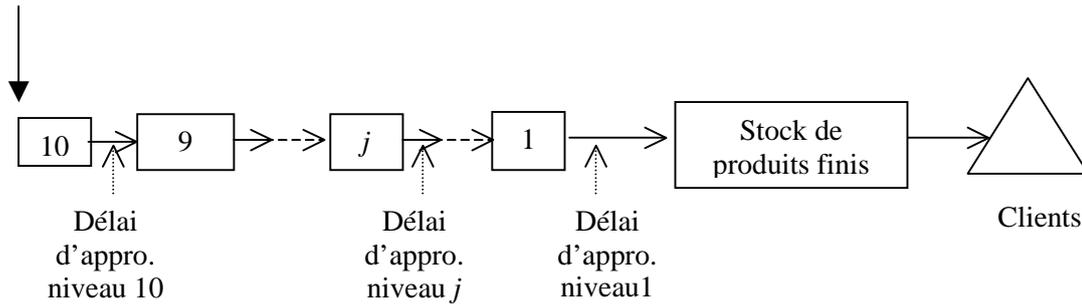
La relation (3.8) n'est vérifiée que lorsque  $b=0$ . Si le coût de rupture est nul alors toute solution  $X_m \leq m - 1$  est optimale et donne le même coût.

C.Q.F.D.

**3.2.1.4 Exemples numériques**

Pour en faire une illustration, nous considérons l'exemple suivant : il s'agit d'une chaîne logistique linéaire à 10 niveaux ( $m=10$ ), les délais d'approvisionnements sont des variables aléatoires discrètes qui suivent une loi quelconque à valeurs bornées tel que la valeur minimale de chaque délai d'approvisionnement est égale à 1 et la valeur maximale des tous les délais d'approvisionnement est  $u=5$ . Le coût unitaire de stockage  $h$  de produits finis est égal à 10 (voir la figure 3.4). La date de livraison voulue  $T=0$ . L'objectif est de déterminer la date de lancement optimale ( $-X_{10}$ ) qui permet de minimiser le coût total moyen.

**Date de lancement  
d'ordre  $T-X_{10}$  ?**



**Figure 3.4 :** Chaîne d'approvisionnement linéaire à 10 niveaux

En premier lieu, nous proposons de tester l'effet du coût de rupture sur la solution optimale. Pour cela, nous avons considéré que tous les délais d'approvisionnement suivent la même loi de distribution et que cette loi est équiprobable=0,2. Le coût de rupture varie entre  $h$  et  $10h$ . Le tableau suivant montre l'augmentation de la solution optimale en fonction de la valeur du coût de rupture.

Ce résultat vient du fait que plus le coût de rupture augmente, plus il faudra éviter la rupture au profit du stockage, et donc lancer le produit plus tôt.

**Tableau 3.1 :** La date optimale de lancement d'ordre de fabrication en fonction de coût de retard

$b =$	$h$	$2h$	$3h$	$4h$	$5h$	$6h$	$7h$	$8h$	$9h$	$10h$
$X_m^*$	30	32	33	34	34	35	35	36	36	36

D'après le tableau 3.1, nous remarquons que parfois  $b$  augmente mais la date de lancement optimale reste la même. Cela vient du fait que même si  $\frac{b}{b+h}$  change la condition d'optimalité :

$$F_{SL}(X_{10}^* - 1) \leq \frac{b}{b+h} \leq F_{SL}(X_{10}^*) \text{ reste vérifiée pour la même valeur de } X_{10}^*.$$

Nous allons également illustrer le comportement de la solution optimale pour différents nombres de niveau de chaîne d'approvisionnement, pour cela nous avons calculé la solution optimale dont la chaîne d'approvisionnement est constituée de 1 niveau, 2 niveaux, ..., jusqu'à 10 niveaux. Le coût de stockage unitaire  $h=10$ , le coût de rupture unitaire  $b=100$  et lorsque tous les délais d'approvisionnement suivent la même loi de distribution et que cette loi est équiprobable = 0,2. le tableau 3.2 représente la date de lancement optimale en fonction de nombre de niveaux.

**Tableau 3.2 :** La date de lancement d'ordre optimale en fonction de nombre de niveaux.

$m$	1	2	3	4	5	6	7	8	9	10
$X_m^*$	5	9	12	16	19	23	26	29	33	36

Nous remarquons que malgré les mêmes caractéristiques des délais d'approvisionnement de tous les niveaux, la date de lancement optimale pour une chaîne d'approvisionnement avec  $j$  niveaux ne peut pas être obtenue avec la formule suivante :

$$X_j \neq j \times X_1, \text{ pour } j=2, \dots, m. \quad (3.9)$$

Si c'était le cas, la date de lancement optimale d'une chaîne à 10 niveaux serait égale à  $50 = 10 \times 5$  ce qui n'est pas le cas. La solution optimale permet de réduire le coût total moyen de 58% par rapport à celle solution.

Enfin, nous avons effectué une étude comparative entre la solution optimale obtenue par les inégalités (3.5) et l'heuristique suivante où la solution considérée est égale à la valeur moyenne  $(T - E(SL))$ . Etant donné que la variable de décision est un entier et que  $E(SL)$  peut être non entier, nous considérons la valeur la plus proche de  $(T - E(SL))$  :  $T - \bar{X}_m$ , tel que  $\bar{X}_m = \lceil E(SL) \rceil$  ou  $\lfloor E(SL) \rfloor$  selon la condition suivante:

$$\bar{X}_m = \begin{cases} \lceil E(SL) \rceil & \text{si } \lfloor E(SL) \rfloor - E(SL) \geq 0.5 \\ \lfloor E(SL) \rfloor & \text{sinon} \end{cases}$$

Nous avons calculé ceci sur 10000 situations générés d'une manière aléatoire (une situation correspond à une loi de distribution de  $L_j$ ,  $j=1, \dots, 10$ ) et les résultats montrent que la solution optimale permet de réduire le coût totale moyen de 21% par rapport à l'utilisation des valeurs moyennes.

### 3.2.2 Problème P2

Nous étudions maintenant le problème lorsque le coût de rupture est remplacé par une contrainte de niveau de service. Le problème est donc la minimisation du coût moyen de stockage de produits finis sous la contrainte d'un niveau de service donné.

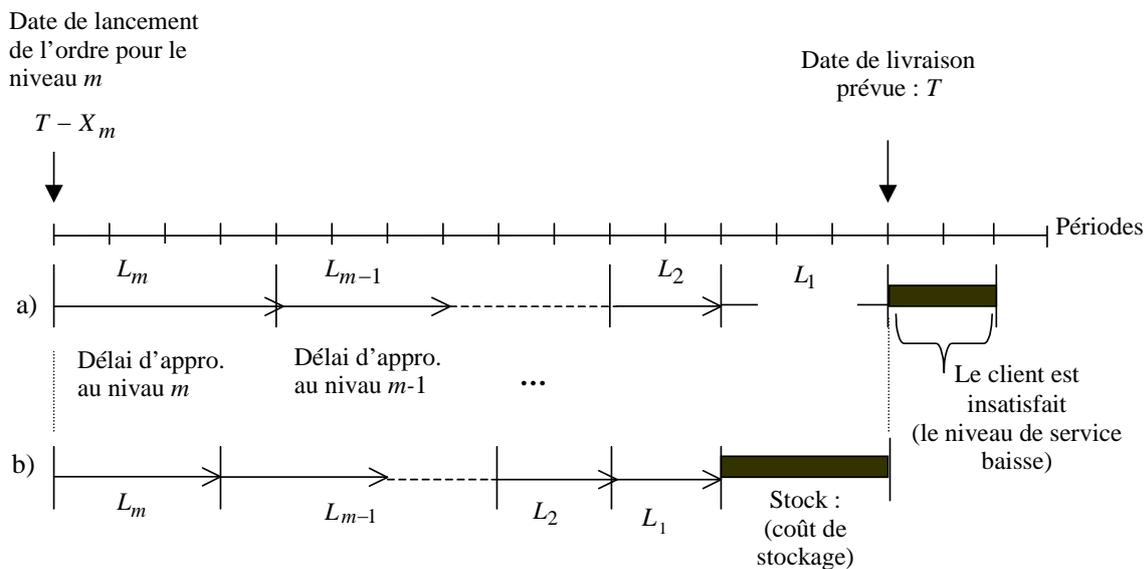
Dans ce que suit, nous présentons analytiquement le coût total moyen à minimiser.

#### 3.2.2.1 Description du problème

En plus des notations précédentes, nous introduisons les notations suivantes :

- $\varepsilon \in [0,1]$  la probabilité de rupture tolérée,
- $1 - \varepsilon$  le niveau de service exigé.

Pareil que pour le problème P1, nous considérons le cas de fabrication au plus tôt : un produit est traité dès qu'il arrive à un nouveau niveau de la chaîne, le problème est de trouver la date de lancement optimale au niveau  $m$  afin de satisfaire la demande  $D$  à la date souhaitée  $T$ , en minimisant le coût total moyen. Si l'ordre est lancé trop tôt, les produits finis arrivent avant la date prévue et donc le stock au niveau 1 augmente, ce qui induit un coût supplémentaire de stockage. Mais, si l'ordre est lancé trop tard, le niveau de service ne sera pas respecté (voir la figure 3.5).



**Figure 3.5** Description du problème P2

### 3.2.2.2 Minimisation des coûts

Nous présentons une modélisation analytique de ce problème. Le critère considéré est la somme du coût moyen de stockage des produits finis. La contrainte est la nécessité de respecter un niveau de service.

**Proposition 3.3**

$$\text{Min } h \sum_{k \geq 0} F_{SL}(X_m - k - 1) \quad (3.10)$$

Sous la contrainte

$$1 - F_{SL}(X_m) \leq \varepsilon \quad (3.11)$$

### Démonstration

La fonction objectif (3.10) représente le coût de stockage des produits finis, cette expression est la première partie de la fonction objectif (3.4). Il reste à développer la contrainte de niveau de service (3.11) :

La probabilité qu'une commande pour les produits finis soit en retard est modélisée comme suit :

$\Pr((L - X_m)^+ > 0)$  , et donc afin de respecter le niveau de service exigé  $(1 - \varepsilon)$ , il faut que :

$$\Pr((L - X_m)^+ > 0) \leq \varepsilon .$$

Alors :  $1 - \Pr((L - X_m)^+ = 0) \leq \varepsilon$

Nous obtenons donc,

$$1 - \Pr(L \leq X_m) \leq \varepsilon .$$

Finalement

$$1 - F_{SL}(X_m) \leq \varepsilon$$

C.Q.F.D.

### **3.2.2.3 Optimisation**

Le coût de stockage (3.10) est une fonction croissante et la probabilité de rupture est une fonction décroissante. La valeur optimale de  $X_m^*$  est donc le plus petit entier qui vérifie la contrainte de niveau de service (3.11).

### **3.2.2.4 Exemples numériques**

Nous considérons l'exemple suivant : il s'agit d'une chaîne logistique linéaire à 10 niveaux ( $m=10$ ), la valeur maximale des délais d'approvisionnement est  $u=5$ . Le coût unitaire de stockage des produits finis est égal à 10 et le niveau de service désiré est de 95% ( $\varepsilon = 0.05$ ). Les délais d'approvisionnement suivent une loi discrète quelconque à valeur bornée entre 1 et  $u$ .

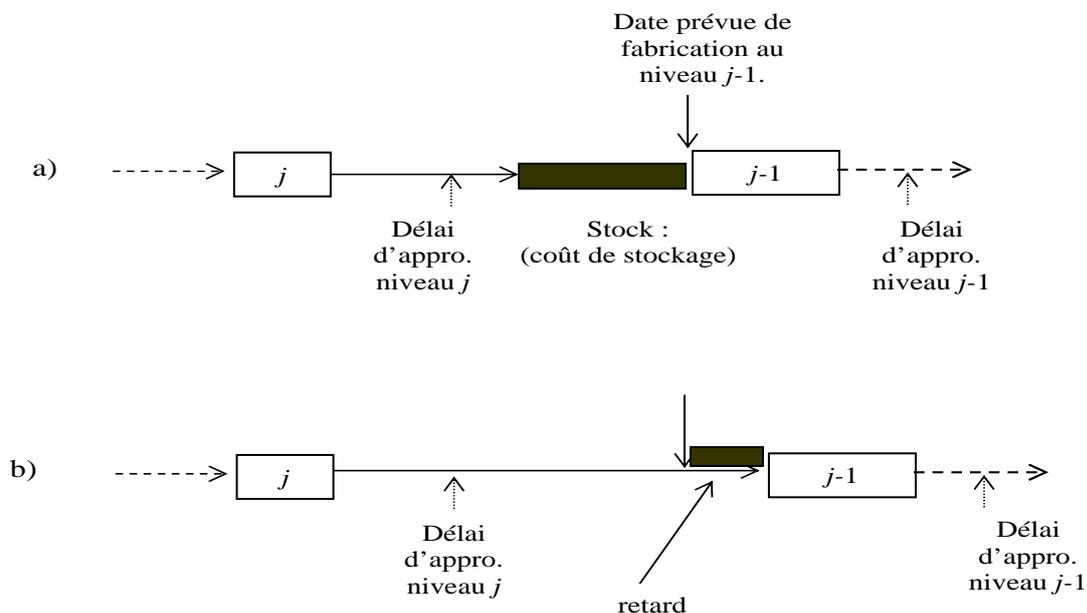
Comme dans l'exemple 1, nous avons effectué une étude comparative entre la solution optimale obtenue par (3.10) – (3.11) et les solutions suivantes :

- La valeur moyenne : la solution entière la plus proche de  $E(SL)$ .
- La solution, notée *modale*, tel que  $X_m = \sum_{j=1}^m x_j$ , où  $x_j$  correspond à la première valeur telle que la fonction de répartition soit supérieure à 0.95 :  $F_j(x_j) \geq 0.95$ .

Nous avons fait une étude avec 10000 scénarios générés d'une manière aléatoire (un scénario correspond à une loi de distribution de  $L_j, j=1, \dots, 10$ ) et les résultats montrent que le niveau de service de la solution utilisant la moyenne est seulement égal à 58 %. Celui de la solution modale est égal à 99%. Mais la solution optimale qui assure un niveau de service de 96% permet de réduire le coût de 70% par rapport à la solution modale.

### 3.3 Le cas avec dates de fabrication

Nous supposons ici que pour assurer la demande en produit fini une entreprise lance la commande chez un fournisseur des matières premières (niveau  $m$ ), le produit commandé traverse toutes les maillons de la chaîne d'approvisionnement pour subir différentes opérations du niveau  $m$  jusqu'au niveau 1. Lorsque le produit est livré de maillon  $j$  au maillon  $j-1$  ( $j=m, m-1, \dots, 2$ ) il ne subit les opérations nécessaires qu'à la date prévue de fabrication. S'il est présent avant cette date il sera stocké le temps restant. Dans le cas contraire, s'il est présent après cette date de fabrication, il subit les opérations nécessaires dès qu'il arrive et ce retard sera pénalisé par un coût de replanification (problème P3) ou bien une contrainte de niveau de service sera imposée à chaque niveau (problème P4) (voir la figure 3.6).



**Figure 3.6 :** Cas avec dates de fabrication

Nous cherchons soit à minimiser le coût moyen de stockage et de replanification à tous les niveaux et le coût moyen de rupture en produit fini (problème P3), soit à minimiser le coût moyen de stockage à tous les niveaux tout en satisfaisant un niveau de service désiré à chaque niveau (problème P4).

Nous présentons la modélisation analytique de chaque problème, les méthodes d'optimisation proposées et des exemples numériques.

### 3.3.1 Problème P3

Du fait de l'existence de dates de fabrication à tous les niveaux, nous introduisons maintenant en plus du coût de rupture et du coût de stockage du produit fini, un coût de stockage des produits semi-finis à tous les niveaux, et des coûts de replanification des niveaux  $m, m-1, \dots, 2$  dû au retard de livraison aux niveaux correspondants. Le problème est donc de trouver la planification optimale qui permet de réduire le coût total moyen. Dans ce qui suit, nous présentons analytiquement le coût total moyen à minimiser.

En plus des notations précédentes, nous introduisons les notations suivantes :

- $h_j$  coût de stockage unitaire du produit de niveau  $j$  ( $j=1, \dots, m$ ) par période,
- $b_1$  coût de rupture unitaire du produit fini par période,
- $b_j$  coût de replanification unitaire du produit de niveau  $j$  ( $j=2, \dots, m$ ) par période.

#### 3.3.1.1 Description du problème P3

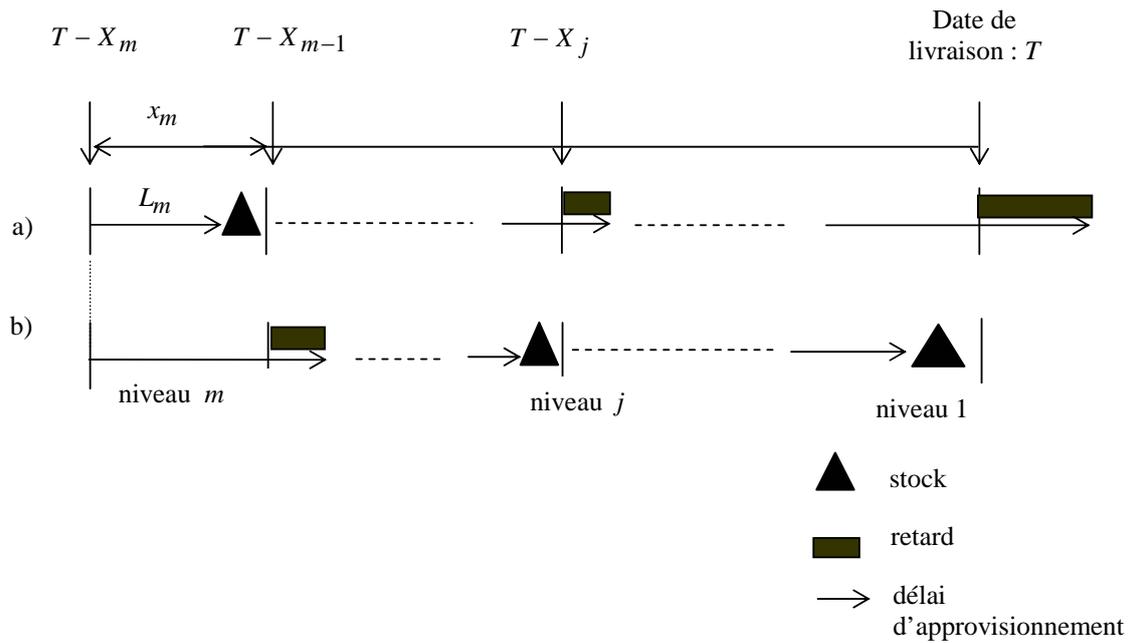
Le problème est donc de trouver les dates de lancement optimales  $T-X_j$  à chaque niveau  $j$  ( $j=1, \dots, m$ ) afin de satisfaire la demande  $D$  (supposée=1) à la date de livraison  $T$  tout en minimisant le coût total (voir la figure 3.7).

Nous supposons que le coût de stockage  $h_j$  de produit de niveau  $j$  ( $j=1, \dots, m$ ), le coût de rupture  $b_1$  en produit fini et les coûts de retard ou de replanification  $b_j$  de niveau  $m, m-1, \dots, 2$  sont connues. Pour les variables aléatoires, nous admettons les mêmes hypothèses que pour les problèmes P1 et P2.

Nous définissons le délai d'approvisionnement cumulé  $L'_j(x_{j+1}, x_{j+2}, \dots, x_m)$  de niveau  $j$  ( $j=1, \dots, m$ ) comme suit :

$$\begin{cases} L'_j(x_{j+1}, x_{j+2}, \dots, x_m) = L_j + (L'_{j+1}(x_{j+2}, x_{j+3}, \dots, x_m) - x_{j+1})^+, \text{ pour } j = 1, 2, \dots, m-1 \\ L'_m = L_m \end{cases} \quad (3.12)$$

où  $x_j$  ( $j=1, \dots, m$ ) est le délai d'approvisionnement planifié. Le délai d'approvisionnement  $L'_j(x_{j+1}, x_{j+2}, \dots, x_m)$ , ( $j=1, \dots, m$ ) est égal donc au délai d'approvisionnement  $L_j$  ( $j=1, \dots, m$ ) plus la somme de retards du niveau  $m, m-1, \dots$ , jusqu'au niveau  $j+1$ .



**Figure 3.7 :** Description du problème P3

Notre objectif est de trouver les dates de lancement des ordres ( $T - X_j$ ) à chaque niveau  $j$  pour  $j = 1, \dots, m$ .

où

$$X_j = \sum_{s=1}^j x_s \quad (3.13)$$

### 3.3.1.2 Minimisation des coûts

Nous présentons le modèle de ce problème en exprimant analytiquement le critère à optimiser. Le critère considéré est la somme des coûts moyens de stockage de tous les niveaux, le coût de rupture en produit fini et le coût de replanification à chaque niveau (du niveau 2 au niveau  $m$ ). Nous proposons une procédure récursive pour trouver la solution optimale. L'objectif est de trouver les dates de lancement optimales à chaque niveau qui minimisent le coût total moyen. D'après (3.13), il suffit donc de trouver les délais d'approvisionnement planifiés optimaux ( $x_j$ ) pour  $j = 1, \dots, m$ .

**Proposition 3.4** Pour le problème P3, le coût total est exprimé comme suit:

$$C(X, L) = \sum_{j=1}^m \left[ h_j (x_j - L'_j(x_{j+1}, x_{j+2}, \dots, x_m)) + (b_j + h_j) (L'_j(x_{j+1}, x_{j+2}, \dots, x_m) - x_j)^+ \right] \quad (3.14)$$

où,

$$X = (x_1, \dots, x_j, \dots, x_m)$$

$$L' = (L'_1, \dots, L'_j, \dots, L'_m)$$

### Démonstration

Le coût total est égal à la somme du coût de stockage des composants à tous les niveaux, du coût de replanification à chaque niveau (du niveau 2 au niveau  $m$ ) et du coût de rupture en produit fini.

Nous allons tout d'abord calculer le coût de stockage :

Si le délai d'approvisionnement réel  $L'_j(x_{j+1}, x_{j+2}, \dots, x_m)$  dépasse le délai d'approvisionnement planifié  $x_j$  :  $(x_j - L'_j(x_{j+1}, x_{j+2}, \dots, x_m)) \geq 0$ , il y a stockage du produit au niveau  $j$ . Le coût de stockage est égal à :  $h_j (x_j - L'_j(x_{j+1}, x_{j+2}, \dots, x_m))^+$  et donc le coût de stockage total est égal à :

$$\sum_{j=1}^m h_j (x_j - L'_j(x_{j+1}, x_{j+2}, \dots, x_m))^+$$

A l'inverse, il y a un retard au niveau  $j$  ( $j=1, \dots, m$ ), si  $(L'_j(x_{j+1}, x_{j+2}, \dots, x_m) - x_j) \geq 0$ , ce qui induit un coût de replanification ou de rupture en produit fini.

La somme du coût de replanification à chaque niveau et du coût de rupture en produit fini est égale à :

$$\sum_{j=1}^m b_j (L'_j(x_{j+1}, x_{j+2}, \dots, x_m) - x_j)^+$$

Donc, le coût total est égal à:

$$C(X, L') = \sum_{j=1}^m b_j (L'_j(x_{j+1}, x_{j+2}, \dots, x_m) - x_j)^+ + \sum_{j=1}^m h_j (x_j - L'_j)^+$$

Etant donné que  $(-f)^+ = \max(-f, 0) = f^- = f^+ - f$  :

Si nous considérons  $f = (x_j - L'_j(x_{j+1}, x_{j+2}, \dots, x_m))$ , le coût total  $C(X, L)$  peut être réécrit comme suit :

$$C(X, L) = \sum_{j=1}^m (b_j + h_j)(L'_j(x_{j+1}, x_{j+2}, \dots, x_m) - x_j)^+ + \sum_{j=1}^m h_j(x_j - L'_j(x_{j+1}, x_{j+2}, \dots, x_m))$$

C.Q.F.D.

Comme le montre la proposition 3.4, le coût total est une variable aléatoire qui est fonction des variables aléatoires  $L'_j(x_{j+1}, x_{j+2}, \dots, x_m)$ ,  $j=1, \dots, m$ .

Nous décidons de nous intéresser au coût moyen  $C(X) = E(C(X, L'))$ .

**Proposition 3.5** Pour le problème P3, le coût total moyen est exprimé comme suit :

$$C(X) = \sum_{j=1}^m \left[ h_j(x_j - E(L'_j(x_{j+1}, x_{j+2}, \dots, x_m))) + (b_j + h_j) \sum_{k \geq 0} (1 - \Phi_j(x_j + k, x_{j+1}, \dots, x_m)) \right] \quad (3.15)$$

où,

$$\begin{cases} \Phi_j(k, x_{j+1}, \dots, x_m) = \sum_{s=1}^k \Pr(L_j = s) \times \Phi_j(x_{j+1} + k - s, x_{j+1}, \dots, x_m), \text{ pour } j = 1, \dots, m-1 \\ \Phi_m(k) = F_m(k) \end{cases} \quad (3.16)$$

$$X = (x_1, \dots, x_j, \dots, x_m)$$

### Démonstration

D'après la relation (3.14), le coût total moyen est le suivant :

$$C(X) = E(C(X, L)) = \sum_{j=1}^m \left[ h_j(x_j - E(L'_j(x_{j+1}, x_{j+2}, \dots, x_m))) + (b_j + h_j)E(Z) \right]$$

où :

$$Z = (L'_j(x_{j+1}, x_{j+2}, \dots, x_m) - x_j)^+.$$

La variable  $Z$  est une variable aléatoire discrète positive avec un nombre fini des valeurs possibles, son espérance mathématique est :

$$E(Z) = \sum_{i \geq 0} i \Pr(Z = i) = \sum_{i \geq 0} \sum_{k=0}^{i-1} \Pr(Z = i) = \sum_{k \geq 0} \sum_{i > k} \Pr(Z = i) = \sum_{k \geq 0} \Pr(Z > k)$$

Nous obtenons donc :

$$\Pr((L'_j(x_{j+1}, x_{j+2}, \dots, x_m) - x_j)^+ > k) = 1 - \Pr((L'_j(x_{j+1}, x_{j+2}, \dots, x_m) - x_j)^+ \leq k)$$

$$= 1 - \left( \Pr((L'_j(x_{j+1}, x_{j+2}, \dots, x_m) - x_j) \leq k) \times \Pr(k \geq 0) \right)$$

$$E(Z) = \sum_{k \geq 0} \left( 1 - \left( \Pr(L'_j(x_{j+1}, x_{j+2}, \dots, x_m) - x_j \leq k) \times \Pr(k \geq 0) \right) \right)$$

Sachant que dans l'expression précédente la somme est calculée pour  $k \geq 0$ , donc :

$$E(Z) = \sum_{k \geq 0} \left( 1 - \Pr(L'_j(x_{j+1}, x_{j+2}, \dots, x_m) \leq x_j + k) \right) = \sum_{k \geq 0} \left( 1 - \Phi_j(x_j + k, x_{j+1}, \dots, x_m) \right)$$

avec,

$$\Phi_j(k, x_{j+1}, \dots, x_m) = \Pr(L'_j(x_{j+1}, x_{j+2}, \dots, x_m) \leq k) = \Pr(L_j + (L'_{j+1}(x_{j+2}, \dots, x_m) - x_{j+1})^+ \leq k)$$

Maintenant, calculons  $\Pr(L_j + (L'_{j+1}(x_{j+2}, \dots, x_m) - x_{j+1})^+ \leq k)$  :

$$\begin{aligned} & \Pr(L_j + (L'_{j+1}(x_{j+2}, \dots, x_m) - x_{j+1})^+ \leq k) \\ &= \sum_{s=1}^k \Pr(L_j = s) \times \Pr\left((L'_{j+1}(x_{j+2}, \dots, x_m) - x_{j+1})^+ \leq k - s\right) \\ &= \sum_{s=1}^k \Pr(L_j = s) \times \Pr\left(L'_{j+1}(x_{j+2}, \dots, x_m) - x_{j+1} \leq k - s\right) \times \Pr(k - s \geq 0) \end{aligned}$$

Sachant que  $k - s$  est toujours positif, donc pour  $j=1, \dots, m-1$  :

$$\Pr(L'_j(x_{j+1}, x_{j+2}, \dots, x_m) \leq k) = \sum_{s=1}^k \Pr(L_j = s) \times \Pr\left(L'_{j+1}(x_{j+2}, \dots, x_m) \leq x_{j+2} + k - s\right)$$

Donc :

$$\Phi_j(k, x_{j+1}, \dots, x_m) = \sum_{s=1}^k \Pr(L_j = s) \times \Phi_j(x_{j+1} + k - s, x_{j+1}, \dots, x_m)$$

Finalement,

$$C(X) = \sum_{j=1}^m \left[ h_j(x_j - E(L'_j(x_{j+1}, x_{j+2}, \dots, x_m))) + (b_j + h_j) \sum_{k \geq 0} (1 - \Phi_j(x_j + k, x_{j+1}, \dots, x_m)) \right]$$

C.Q.F.D.

La fonction objectif (3.15) est non linéaire à variables de décision entières. La résolution de ce problème nécessite une étude des accroissement partiels  $G_j^+(X)$  et  $G_j^-(X)$  de la fonction objectif  $C(X)$ .

### 3.3.1.3 Optimisation

#### 3.3.1.3.1 Espace de recherche des variables de décision

**Proposition 3.6** Les valeurs optimales des variables de décision sont dans l'espace de recherche de solution suivant :

$$1 \leq x_j \leq u_j + \sum_{s=j+1}^m (u_s - 1), \text{ pour } j=1, \dots, m-1 \quad (3.17)$$

$$1 \leq x_m \leq u_m \quad (3.18)$$

#### Démonstration

Nous commençons par démontrer l'inégalité (3.18) :

Étant donné que le délai d'approvisionnement de niveau  $m$  varie entre 1 et  $u_m$  alors :

Pour tous  $x_m > u_m$ ,  $F_m(x_m) = 1$ .

Donc pour :  $s > 0$ ,  $y = x_m + s$  et  $x_m = u_m$  :

$$C(x_1, \dots, x_{m-1}, y) - C(x_1, \dots, x_m) = s \times h_m \geq 0.$$

Ce qui prouve que pour tous  $x_m > u_m$ , nous obtenons un coût de stockage supplémentaire.

La propriété (3.18) et donc démontrée.

Pour démontrer (3.17), le principe est le même :

Montrer que :  $\Phi_j(x_j, x_{j+1}, \dots, x_m) = 1$  lorsque  $x_j > u_j + \sum_{s=j+1}^m (u_s - 1)$

Alors, nous obtenons un coût de stockage supplémentaire pour tous  $x_j > u_j + \sum_{s=j+1}^m (u_s - 1)$

C.Q.F.D.

#### 3.3.1.3.2 Propriétés des variables de décision

Dans cette section, nous allons présenter des propriétés des variables de décision qui nous permettront ensuite de réaliser une procédure de résolution optimale du problème P3.

Par analogie avec la thèse de Ould-Louly (2001), nous introduisons les accroissements partiels à droite et à gauche par rapport à la variable  $x_j$  qui sont les fonctions suivantes :

$$G_j^+(X) = C(x_1, \dots, x_j + 1, \dots, x_m) - C(x_1, \dots, x_j, \dots, x_m) \quad (3.19)$$

$$G_j^-(X) = C(x_1, \dots, x_j - 1, \dots, x_m) - C(x_1, \dots, x_j, \dots, x_m) \quad (3.20)$$

D'après la définition de  $G_j^+(X)$  et  $G_j^-(X)$ , nous pouvons dire que les expressions suivantes sont des conditions nécessaires pour que le vecteur  $X$  donne le minimum de la fonction  $C(X)$  :

$$G_j^+(X) \geq 0, \text{ pour } j = 1, \dots, m \quad (3.21)$$

$$G_j^-(X) \geq 0, \text{ pour } j = 1, \dots, m \quad (3.22)$$

**Proposition 3.7** La fonction  $G_j^+(X)$  vérifie les propriétés suivantes :

$$G_j^+(X) \leq \left[ -b_j + (b_j + h_j)\Phi_j(x_j, x_{j+1}, \dots, x_m) \right] + \sum_{s=1}^{j-1} h_s \quad (3.23)$$

$$G_j^+(X) \geq \left[ -b_j + (b_j + h_j)\Phi_j(x_j, x_{j+1}, \dots, x_m) \right] - \sum_{s=1}^{j-1} (b_s + h_s) \quad (3.24)$$

### Démonstration

$$G_j^+(X) = \sum_{s=1}^m \left[ h_s(x_s - E(L'_s(x_{s+1}, \dots, x_j + 1, \dots, x_m))) + (b_s + h_s) \sum_{k \geq 0} (1 - \Phi_s(x_s + k, \dots, x_j + 1, \dots, x_m)) \right] \\ - \sum_{s=1}^m \left[ h_s(x_s - E(L'_s(x_{s+1}, \dots, x_j, \dots, x_m))) + (b_s + h_s) \sum_{k \geq 0} (1 - \Phi_s(x_s + k, \dots, x_j, \dots, x_m)) \right]$$

La différence entre ces deux coûts peut être calculée terme par terme selon la valeur de  $s$ . Les termes peuvent être séparés pour faciliter l'explication. Soit  $G_j^+(X) = A + B + C$ .

Le premier terme est pour les valeurs de  $s$  supérieur à  $j$ . Il est égal à zéro :

$$A = \sum_{s=j+1}^m \left[ h_s(x_s - E(L'_s(x_{s+1}, \dots, x_m))) + (b_s + h_s) \sum_{k \geq 0} (1 - \Phi_s(x_s + k, \dots, x_m)) \right] \\ - \sum_{s=j+1}^m \left[ h_s(x_s - E(L'_s(x_{s+1}, \dots, x_m))) + (b_s + h_s) \sum_{k \geq 0} (1 - \Phi_s(x_s + k, \dots, x_m)) \right] = 0$$

Le deuxième terme est pour  $s = j$ . Il se calcule de la manière suivante :

$$B = \left[ h_j(x_{j+1} - E(L'_j(x_{j+1}, x_{j+2}, \dots, x_m))) + (b_j + h_j) \sum_{k \geq 0} (1 - \Phi_j(x_{j+1} + k, x_{j+1}, \dots, x_m)) \right] \\ - \left[ h_j(x_j - E(L'_j(x_{j+1}, x_{j+2}, \dots, x_m))) + (b_j + h_j) \sum_{k \geq 0} (1 - \Phi_j(x_j + k, x_{j+1}, \dots, x_m)) \right] \\ = [h_j - (b_j + h_j)(1 - \Phi_j(x_j, x_{j+1}, \dots, x_m))]$$

$$=[-b_j + (b_j + h_j)\Phi_j(x_j, x_{j+1}, \dots, x_m)]$$

Le troisième terme est pour  $s < j$  :

$$\begin{aligned} C &= \sum_{s=1}^{j-1} \left[ h_s (x_s - E(L'_s(x_{s+1}, \dots, x_j + 1, \dots, x_m))) + (b_s + h_s) \sum_{k \geq 0} (1 - \Phi_s(x_s + k, \dots, x_j + 1, \dots, x_m)) \right] \\ &\quad - \sum_{s=1}^{j-1} \left[ h_s (x_s - E(L'_s(x_{s+1}, \dots, x_j, \dots, x_m))) + (b_s + h_s) \sum_{k \geq 0} (1 - \Phi_s(x_s + k, \dots, x_j, \dots, x_m)) \right] \\ &= \sum_{s=1}^{j-1} h_s \left[ E(L'_s(x_{s+1}, \dots, x_j, \dots, x_m)) - E(L'_s(x_{s+1}, \dots, x_j + 1, \dots, x_m)) \right] \\ &\quad + \sum_{s=1}^{j-1} (b_s + h_s) \left[ \sum_{k \geq 0} (1 - \Phi_s(x_s + k, \dots, x_j + 1, \dots, x_m)) - \sum_{k \geq 0} (1 - \Phi_s(x_s + k, \dots, x_j, \dots, x_m)) \right] \end{aligned}$$

$$\text{D'autre part, } 0 \leq E(L'_s(x_{s+1}, \dots, x_j, \dots, x_m)) - E(L'_s(x_{s+1}, \dots, x_j + 1, \dots, x_m)) \leq 1$$

$$\text{et } 0 \leq \sum_{k \geq 0} (1 - \Phi_s(x_s + k, \dots, x_j, \dots, x_m)) - \sum_{k \geq 0} (1 - \Phi_s(x_s + k, \dots, x_j + 1, \dots, x_m)) \leq 1$$

$$\text{Alors, le terme } C \text{ vérifie l'inégalité suivante: } -\sum_{s=1}^{j-1} (b_s + h_s) \leq C \leq \sum_{s=1}^{j-1} h_s$$

Finalement, nous obtenons :

$$\begin{aligned} G_j^+(X) &\leq [-b_j + (b_j + h_j)\Phi_j(x_j, x_{j+1}, \dots, x_m)] + \sum_{s=1}^{j-1} h_s \\ G_j^+(X) &\geq [-b_j + (b_j + h_j)\Phi_j(x_j, x_{j+1}, \dots, x_m)] - \sum_{s=1}^{j-1} (b_s + h_s) \end{aligned}$$

C.Q.F.D.

**Proposition 3.8** Les expressions (3.23) et (3.24) nous permettent d'obtenir les propriétés des variables de décision suivantes :

$$\bullet \Phi_j(x_j, x_{j+1}, \dots, x_m) \geq \alpha_j \quad \text{pour } j=1, \dots, m \quad (3.25)$$

$$\bullet \Phi_j(x_j - 1, x_{j+1}, \dots, x_m) \leq \beta_j \quad \text{pour } j=1, \dots, m \quad (3.26)$$

$$\bullet F_m(x_m) \geq \alpha_m \quad (3.27)$$

$$\bullet F_m(x_m - 1) \leq \beta_m \quad (3.28)$$

$$\bullet F_j(x_j) \geq \alpha_j \quad \text{pour } j=1, \dots, m \quad (3.29)$$

$$\text{Où } \alpha_j = \frac{b_j - \sum_{s=1}^{j-1} h_s}{b_j + h_j} \text{ et } \beta_j = \frac{b_j + \sum_{s=1}^{j-1} (b_s + h_s)}{b_j + h_j}, \text{ pour } j=1, \dots, m$$

### Démonstration

Selon la propriété (3.23) :  $G_j^+(X) \leq \left[ -b_j + (b_j + h_j)\Phi_j(x_j, x_{j+1}, \dots, x_m) \right] + \sum_{s=1}^{j-1} h_s$

Donc :

$$\frac{b_j - \sum_{s=1}^{j-1} h_s}{b_j + h_j} \leq \Phi_j(x_j, x_{j+1}, \dots, x_m), \text{ pour } j=1, \dots, m.$$

Donc, la propriété (3.25) est démontrée.

Comme  $G_j^-(X) = -G_j^+(x_1, \dots, x_j - 1, \dots, x_m)$ , nous obtenons l'inégalité suivante de (3.22):

$$0 \leq \left[ b_j - (b_j + h_j)\Phi_j(x_j - 1, x_{j+1}, \dots, x_m) \right] + \sum_{s=1}^{j-1} (b_s + h_s)$$

Donc,

$$\Phi_j(x_j - 1, x_{j+1}, \dots, x_m) \leq \frac{b_j + \sum_{s=1}^{j-1} (b_s + h_s)}{b_j + h_j}.$$

Donc, la propriété (3.26) est démontrée.

En utilisant la définition (3.16), la propriété (3.27) est immédiatement dérivée de (3.25) et la propriété (3.28) est immédiatement dérivée de (3.26).

En utilisant la définition (3.16) :

$$\Phi_j(x_j, x_{j+1}, \dots, x_m) \leq F_j(x_j), \text{ pour } j=1, \dots, m.$$

Finalement,

$$\frac{b_j - \sum_{s=1}^{j-1} h_s}{b_j + h_j} \leq F_j(x_j), \text{ pour } j=1, \dots, m$$

Nous obtenons la propriété (3.29).

C.Q.F.D.

Les équations (3.25)-(3.29), nous permettent d'obtenir des limites inférieures des variables de décision et donc de réduire l'espace de recherche des solutions.

Nous notons par  $LL = (ll_1, \dots, ll_j, \dots, ll_m)$  le vecteur de limites inférieures des variables de décision  $(x_1, \dots, x_j, \dots, x_m)$ . Le vecteur des limites supérieures des variables de décision est notée par  $UL = (ul_1, \dots, ul_j, \dots, ul_m)$ .

### 3.3.1.3.3 Procédure récursive de réduction d'espace de recherche

En se basant sur ces propriétés des variables de décision, nous développons une procédure récursive. Cette procédure permet de trouver la solution pour un nombre quelconque de niveaux dans la chaîne logistique en temps de calcul raisonnable.

Nous commençons la procédure par la recherche de valeurs possibles de délai d'approvisionnement optimale pour le niveau  $m$  noté par  $x_m$  en utilisant (3.27) et (3.28).

Donc :

$$F^{-1}(\alpha_m) \leq x_m \leq F^{-1}(\beta_m) + 1 \quad (3.30)$$

Ensuite, en utilisant (3.25), (3.26) et (3.29), nous pouvons développer une procédure récursive qui permet d'obtenir les valeurs possibles des délais d'approvisionnement  $(x_1, \dots, x_{m-1})$ .

Donc, la procédure qui nous permet d'obtenir l'espace des solutions possibles est la suivante :

#### Procédure récursive pour réduire l'espace de recherche

##### Début

// Limites inférieure et supérieure pour le variable du niveau  $m$  :

$$ll_m \leftarrow \text{le plus petit entier } i \text{ qui vérifie : } \alpha_m \leq F_m(i)$$

$$ul_m \leftarrow \text{le plus grand entier } i \text{ qui vérifie: } F_m(i-1) \leq \beta_m$$

//Initialisation des limites inférieures et supérieures de tous les autres variables :

$$ll_j \leftarrow 1 \quad \text{pour } j = 1 \text{ to } m-1$$

$$ul_j \leftarrow u_j + \sum_{s=j+1}^m (u_s - 1) \quad \text{pour } j = 1 \text{ to } m-1$$

// Limites inférieures et supérieures des variables des niveaux de  $m-1$  à 1 :

**Pour**  $j = m-1$  à 1 faire

// Limites supérieures des variables :

$$Y \leftarrow ll_j$$

**Tant que**  $\Phi_j(Y-1, ll_{j+1}, \dots, ll_m) \leq \beta_j$  faire

$ul_j \leftarrow Y$

$Y \leftarrow Y+1$

**Fin Tant que**

*// Limites inférieures des variables :*

$Y \leftarrow ul_j$

**Tant que**  $\alpha_j \leq F_j(Y)$  **ou**  $\alpha_j \leq \Phi_j(Y, ul_{j+1}, \dots, ul_m)$  faire

$ll_j \leftarrow Y$

$Y \leftarrow Y-1$

**Fin Tant que**

**Fin pour**

**Fin**

### 3.3.1.3.4 Borne inférieure de la fonction objectif

Nous allons maintenant présenter une borne inférieure de la fonction objectif que nous utilisons dans notre algorithme d'optimisation.

**Proposition 3.9** *La valeur BI suivante est une borne inférieure de la fonction objectif du problème P3 dans l'espace  $[LL, UL]$  :*

$$BI = \sum_{j=1}^m \left[ h_j (ll_j - E(L'_j(ll_{j+1}, ll_{j+2}, \dots, ll_m))) + (b_j + h_j) \sum_{k \geq 0} (1 - \Phi_j(ul_j + k, ul_{j+1}, \dots, ul_m)) \right] \quad (3.31)$$

Démonstration

$E(L'_j(x_{j+1}, x_{j+2}, \dots, x_m))$  est une fonction décroissante par rapport à  $x_s$  pour  $s=j+1, \dots, m$ , et

$\Phi_j(x_j + k, x_{j+1}, \dots, x_m)$  est une fonction croissante par rapport à  $x_s$  pour  $s=j+1, \dots, m$ .

Alors, pour tout élément  $x_j, j=1, \dots, m$ , de  $[LL, UL]$  :

$$\begin{aligned} C(X) &= \sum_{j=1}^m \left[ h_j (x_j - E(L'_j(x_{j+1}, x_{j+2}, \dots, x_m))) + (b_j + h_j) \sum_{k \geq 0} (1 - \Phi_j(x_j + k, x_{j+1}, \dots, x_m)) \right] \\ &\geq \sum_{j=1}^m \left[ h_j (ll_j - E(L'_j(ll_{j+1}, ll_{j+2}, \dots, ll_m))) + (b_j + h_j) \sum_{k \geq 0} (1 - \Phi_j(ul_j + k, ul_{j+1}, \dots, ul_m)) \right] \end{aligned}$$

C.Q.F.D.

La procédure récursive de réduction d'espace de recherche et la borne inférieure de la fonction objectif donnent la possibilité de développer une procédure d'optimisation exacte. Nous proposons une Procédure par Séparation et Evaluation (PSE). Cet algorithme s'est inspiré de l'algorithme analogue dans la thèse de Ould Louly (2001).

L'espace de recherche initial des solutions est noté par  $[LL, UL]$ , où  $LL$  représente les limites inférieures des variables de décision et  $UL$  représente les limites supérieures des variables de décision :

$$ll_j \leq x_j \leq ul_j, \quad j=1, \dots, m.$$

Après la division d'un nœud, dans une PSE, deux nœuds (descendants) sont créés. Les descendants du nœud  $[A, B]$  représentant l'espace de recherche avec  $A=(a_1, a_2, \dots, a_j, \dots, a_m)$  et  $B=(b_1, b_2, \dots, b_j, \dots, b_m)$  où initialement  $a_j=ll_j$  et  $b_j=ul_j$ , sont obtenus en partageant cet ensemble en deux sous-ensembles  $[A, B^1]$  et  $[A^1, B]$  respectivement  $a_j \leq x_j \leq \frac{a_j + b_j}{2}$  et  $\frac{a_j + b_j}{2} + 1 \leq x_j \leq b_j$  suivant la sous ensemble  $[a_j, b_j]$  de cardinalité maximale.

### 3.3.1.3.5 Borne supérieure

Nous avons choisi comme borne supérieure (BS) initiale dans la PSE le résultat de la procédure suivante :

#### Procédure Borne supérieure

Cette procédure commence par l'initialisation de l'espace de recherche

$$A \leftarrow LL$$

$$B \leftarrow UL$$

Ensuite, la procédure suivante est appliquée :

$$BS \leftarrow \min(C(A), C(B))$$

Tant que (le cardinal de  $[A, B]$  est supérieur à 1) faire :

- Séparer  $[A, B]$  en deux sous ensembles  $[A, B^1]$  et  $[A^1, B]$  (comme nous l'avons mentionné ci-dessus cette division du pavé  $[A, B]$  en deux pavés  $[A, B^1]$  et  $[A^1, B]$  est faite en coupant le coté le plus long du pavé  $[A, B]$ ).
- $BS \leftarrow$  la solution avec le coût le plus faible, choisie parmi les quatre possibilités :  $A, B^1, A^1$  ou  $B$ .
- Si  $A$  ou  $B^1$  alors

$$[A, B] \leftarrow [A, B^1]$$

sinon

$[A, B] \leftarrow [A^1, B]$

Fin Si

Fin Tant que

### Algorithme PSE

L'algorithme PSE que nous proposons utilise pour le branchement le schéma suivant. Le sous-ensemble de pavé  $[A, B]$  dont le cardinal est le plus grand est divisé en deux sous-ensembles fils  $[A, B^1]$  et  $[A^1, B]$ . Chaque fils est exploré, la recherche est donc en largeur d'abord.

### Début

// Réduction d'espace de recherche des solutions :

Procédure récursive de réduction de l'espace initial :

$A \leftarrow LL$

$B \leftarrow UL$

$BS \leftarrow \max(C(A), C(B))$

$Le\_min\_actuel \leftarrow BS$

$La\_solution\_actuelle \leftarrow$  la solution donnée par  $BS$

// Initialisation de l'ensemble  $E$  des sous-ensembles :

$E \leftarrow \{[A, B]\}$

**Si**  $A \neq B$  alors

**Tant que** ( $E$  n'est pas vide) faire

$[A, B] \leftarrow$  Prendre l'élément de  $E$  de cardinalité maximale

$E \leftarrow E - \{[A, B]\}$

Diviser  $[A, B]$  en deux sous-ensembles :  $[A, B^1]$  et  $[A^1, B]$

**Si**  $(BI([A, B^1]) < Le\_min\_actuel)$  alors

On l'ajoute à  $E$  :  $E \leftarrow E \cup \{[A, B^1]\}$

**Fin si**

**Si**  $(BI([A, B^1]) < Le\_min\_actuel)$  alors

On l'ajoute à  $E : E \leftarrow E \cup \{[A^1, B]\}$

**Fin si**

// Réactualisation de tous les sous-ensembles si nécessaire :

$BS \leftarrow \min(C(A^1), C(B^1))$

**Si** ( $Le\_min\_actuel > BS$ ) alors

$Le\_min\_actuel \leftarrow BS$

$La\_solution\_actuelle \leftarrow \operatorname{argmin}(C(A^1), C(B^1))$

Eliminer les éléments de  $E$  dont la borne inférieure (BI) est plus grande que le  $Le\_min\_actuel$

**Fin si**

**Fin Tant que**

**Fin si**

**Fin**

### 3.3.1.4 Exemples numériques

Pour illustrer la procédure récursive pour la réduction de l'espace de recherche pour le problème P3, nous considérons l'exemple suivant : il s'agit d'une chaîne logistique linéaire à 5 niveaux ( $m=5$ ), la valeur maximale des délais d'approvisionnement est  $u=5$ . Le coût de stockage, le coût de replanification et le coût de rupture sont présentés dans le tableau 3.3. La loi de probabilité des délais d'approvisionnement est présentée dans le tableau 3.4 (a été générée aléatoirement).

**Tableau 3.3 :** Coût de stockage, de replanification et de rupture

$j$	1	2	3	4	5
$h_j$	54	18	6	2	1
$b_j$	434	142	48	18	9

**Tableau 3.4 :** Loi de probabilité des délais d'approvisionnement

$w$	1	2	3	4	5
$\Pr(L_1=w)$	0.067982	0.335383	0.277422	0.200515	0.118698
$\Pr(L_2=w)$	0.019112	0.116578	0.464832	0.187887	0.211592
$\Pr(L_3=w)$	0.630887	0.284445	0.076000	0.002980	0.005687
$\Pr(L_4=w)$	0.140494	0.197669	0.212363	0.223733	0.225741
$\Pr(L_5=w)$	0.098408	0.392513	0.266861	0.208452	0.033766

En utilisant la procédure récursive de solutions possibles nous réduisons l'espace de recherche des solutions comme suit :

**Tableau 3.5 :** Les limites inférieures et supérieures

$j$	1	2	3	4	5
$ll_j$	4	3	1	1	1
$ul_j$	21	17	13	9	5

La table 3.6 montre que la cardinalité d'espace de recherche des solutions est réduite de  $\prod_{j=0}^{m-1} (u_{m-j} - j) = 208845$  à 157950 éléments. C'est 24% de moins par rapport à l'espace initiale. Ce

pourcentage de réduction dépend des valeurs des coûts unitaires de stockage et des coûts unitaires de retard de produit et des valeurs de la loi de distribution des délais d'approvisionnement.

La solution optimale de cet exemple est présentée dans le tableau suivant :

**Tableau 3.6 :** La solution optimale

$j$	1	2	3	4	5
$x_j^*$	5	5	5	5	5

Nous avons testé la procédure de réduction de l'espace sur 10000 scénarios générés d'une manière aléatoire (un scénario correspond à une loi de distribution de  $L_j$ ,  $j=1, \dots, 5$ , coût de retard et de stockage à chaque niveau). Les résultats montrent que la procédure récursive permet de réduire l'espace de recherche, en moyenne, de 19,6% par rapport à l'espace initiale.

Nous avons également effectué une étude comparative entre les solutions suivantes : i) optimale et ii) moyenne (la solution la plus proche du délai d'approvisionnement moyen à chaque niveau) et iii) obtenue en appliquant le modèle du marchand de journaux à chaque niveau. Nous avons considéré le cas d'une chaîne d'approvisionnement à 5 niveaux. Les résultats, pour 100 instances générées aléatoirement, montrent que la PSE permet de réduire le coût, en moyenne, de 85% par rapport à la solution moyenne et de 52% par rapport à la solution en appliquant le modèle du marchand de journaux niveau par niveau.

### 3.3.2 Problème P4

#### 3.3.2.1 Description du problème

Nous intéressons maintenant au problème P4 : la minimisation du coût de stockage sous les contraintes de niveau de service au client final et de taux de retards à chaque niveau.

En plus des notations précédentes, nous introduisons les notations suivantes :

- $\varepsilon_j \in [0,1]$  la probabilité de rupture tolérée au niveau  $j$  ( $j=1,\dots,m$ )
- $1-\varepsilon_j$  le niveau de service exigé au niveau  $j$  ( $j=1,\dots,m$ )

Du fait de l'existence de dates de fabrication à tous les niveaux, nous introduisons maintenant en plus du coût de stockage du produit à chaque niveau, un taux de retard à chaque niveau. Si l'ordre est lancé après la date de fabrication cela est considéré comme un retard (voir la figure 3.8). Le problème est donc de trouver la planification optimale qui permet de réduire le coût total moyen tout en respectant le niveau de service (taux de retards) à tous les niveaux. Dans ce qui suit, nous présentons analytiquement le coût total moyen à minimiser et les contraintes de niveau de service à respecter.

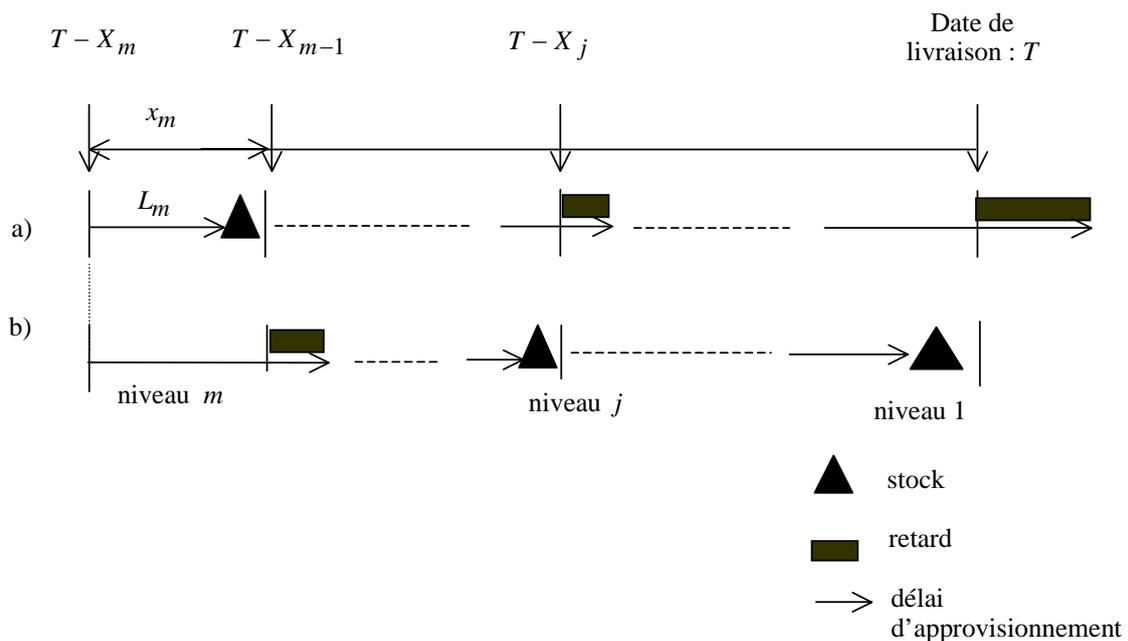


Figure 3.8 : Description du problème P4

### 3.3.2.2 Minimisation du coût moyen

**Proposition 3.10** *Le problème P4 est modélisé comme suit*

*Minimiser le coût :*

$$C(X) = \sum_{j=1}^m h_j \sum_{k \geq 0} (\Phi_j(x_j - k - 1, x_{j+1}, \dots, x_m)) \quad (3.31)$$

*Sous les contraintes suivantes :*

$$\Phi_j(x_{j+1}, x_{j+2}, \dots, x_m) \geq 1 - \varepsilon_j, \quad \forall j = 1, \dots, m. \quad (3.32)$$

*où,*

$$\begin{cases} \Phi_j(k, x_{j+1}, \dots, x_m) = \sum_{s=1}^k \Pr(L_j = s) \times \Phi_j(x_{j+1} + k - s, x_{j+1}, \dots, x_m), \text{ pour } j = 1, \dots, m-1 \\ \Phi_m(k) = F_m(k) \end{cases} \quad (3.33)$$

$\varepsilon_j \in [0, 1]$  est la probabilité de rupture tolérée au niveau  $j$ . Le taux de service exigé est

alors égal à  $1 - \varepsilon_j$ .

#### Démonstration

La fonction objectif (3.31) représente le coût de stockage, elle a été prouvée dans (3.15).

La probabilité que le produit du niveau  $j$  soit en retard est modélisée comme suit :

$$\Pr((L'_j(x_{j+1}, x_{j+2}, \dots, x_m) - x_j)^+ > 0), \quad \forall j = 1, \dots, m.$$

Donc afin de respecter le niveau de service exigé ( $1 - \varepsilon_j$ ), il faut que la probabilité de rupture obtenue soit inférieure à la probabilité de rupture tolérée :

$$\Pr((L'_j(x_{j+1}, x_{j+2}, \dots, x_m) - x_j)^+ > 0) \leq \varepsilon_j, \quad \forall j = 1, \dots, m \quad (3.34)$$

Nous avons donc,

$$1 - \Pr((L'_j(x_{j+1}, x_{j+2}, \dots, x_m) - x_j)^+ = 0) \leq \varepsilon_j, \quad \forall j = 1, \dots, m$$

Donc :

$$1 - \Pr(L'_j(x_{j+1}, x_{j+2}, \dots, x_m) \leq x_j) \leq \varepsilon_j$$

Finalement :

$$\Phi_j(x_j, x_{j+1}, x_{j+2}, \dots, x_m) \geq 1 - \varepsilon_j, \quad \forall j = 1, \dots, m$$

C.Q.F.D.

### 3.3.2.3 Optimisation

Sachant que le coût de stockage (3.29) est une fonction croissante de  $x_j$ , la valeur optimale  $x_j^*$  ( $j=1, \dots, m$ ) est donc le plus petit entier qui vérifie la contrainte (3.30). La première étape consiste à trouver  $x_m^*$ , le plus petit entier qui vérifie  $F_m(x_m) \geq 1 - \varepsilon_m$ , et récursivement  $x_{m-1}^*$  la plus petite valeur qui satisfait  $\Phi_{m-1}(x_j) \geq 1 - \varepsilon_{m-1}, \dots$ , et ainsi de suite jusqu'à  $x_1^*$ .

### 3.3.2.4 Tests numériques

Nous avons considéré l'exemple suivant : il s'agit d'une chaîne logistique linéaire à 5 niveaux ( $m=5$ ), la valeur maximale des délais d'approvisionnement est  $u=5$ . Le taux de service désiré à chaque niveau est de 95% ( $\varepsilon=0.05$ ). Les délais d'approvisionnement suivent une loi discrète quelconque à valeur bornée entre 1 et  $u$ .

Nous avons effectué une étude comparative entre la solution optimale et la solution, notée *modale*, telle que  $x_j$  correspond à la première valeur quand la fonction de répartition devient supérieure à 0.95 :  $F_j(x_j) \geq 0.95$ .

Nous avons testé cette étude sur 100 scénarios générés d'une manière aléatoire (un scénario correspond à une loi de distribution de  $L_j, j=1, \dots, 5$ , et un coût de stockage à chaque niveau) et les résultats montrent que la solution modale permet d'obtenir la solution optimale pour cet exemple et ce niveau de service de 0.95%.

Nous avons testé l'effet du niveau de service exigé sur l'écart entre le coût de la solution modale et le coût de la solution optimale. Les résultats montrent que l'écart varie entre 100% et 0% pour un niveau de service qui varie respectivement entre 10% à 95%.

## 3.4 Conclusion

Nous nous sommes intéressés dans ce chapitre aux chaînes logistiques linéaires. Ceci revient à étudier le parcours d'un produit « du fournisseur de matières première au client final ». Nous avons alors différencié deux cas : soit le produit est traité dès qu'il arrive (le règle juste à temps), soit une date de livraison planifiée existe pour chaque étape. Enfin, nous avons tout d'abord modélisé ces problèmes avec un coût de rupture. Or, dans la réalité, ce coût de rupture est souvent difficile à estimer, et il est alors remplacé par un niveau de service. Nous avons donc également étudié une modélisation de ces problèmes avec une contrainte de niveau de service.

Pour tous ces cas, une modélisation a été faite, des propriétés des modèles ont été démontrées et une procédure d'obtention de la solution optimale est proposée, sachant que notre critère d'optimisation est le coût total moyen.

Tout notre travail est illustré par des exemples numériques commentés.

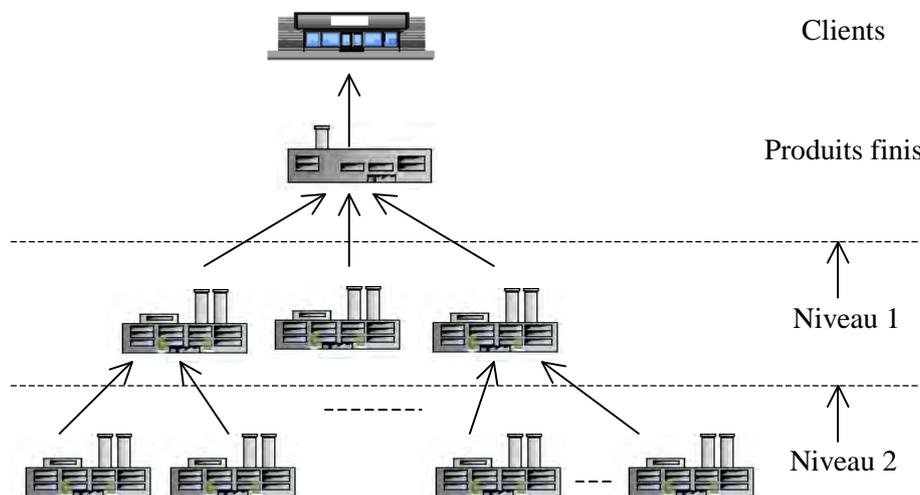
Dans les deux chapitres suivants, nous nous intéresserons au même problème, mais pour une chaîne d'approvisionnement de structure différente ; il s'agit à étudier des chaînes d'assemblage. Ce problème est plus complexe car en plus de la dépendance entre les niveaux, il y a également la dépendance entre les délais d'approvisionnement des composants nécessaires pour assembler un produit, puisque l'assemblage ne peut se faire tant que tous les composants nécessaires ne soient pas présents.

Dans le chapitre suivant, nous présentons une modélisation détaillée et une procédure optimale basée sur la séparation et évaluation. Le chapitre 5 présente une approche heuristique basée sur un algorithme génétique afin de résoudre des problèmes de grande taille.

# Chapitre 4 : Optimisation des systèmes d'assemblage à deux niveaux par une PSE

## 4.1 Introduction

Après avoir présenté dans le chapitre précédent une technique qui permet d'optimiser la planification d'approvisionnement au sein d'une chaîne linéaire d'approvisionnement, nous étudions dans ce chapitre le problème de la planification des réapprovisionnements des systèmes d'assemblage. Un système d'assemblage a une spécificité liée avec les flux convergents : Le produit fini est obtenu par l'assemblage de plusieurs composants, ceux-ci étant eux-mêmes obtenus par l'assemblage d'autres composants.



**Figure 4.1** : Exemple d'un système d'assemblage à deux niveaux

Nous nous intéressons à la planification des réapprovisionnements d'un système d'assemblage à deux niveaux quand les délais d'approvisionnement en composants à tous les niveaux sont des variables aléatoires discrètes qui suivent des lois quelconques et connues et quand la demande en produit fini et sa date de livraison sont supposés connus.

L'objectif est de trouver les valeurs optimales des dates de lancement des ordres de niveau 2 en étudiant deux critères : la minimisation de la somme du coût moyen de stockage des composants et du coût moyen de rupture en produit fini (problème PP1) ou la minimisation du coût moyen de stockage des composants sous la contrainte de niveau de service (problème PP2).

La description générale du système étudié est décrite dans la section suivante. Dans la section 4.3 le problème PP1 est étudié et dans la section 4.4 le problème PP2 est étudié. Nous développons dans chacune de ces deux sections la fonction objectif à minimiser par séparation et évaluation PSE qui permet de trouver pour ces problèmes les valeurs optimales des dates de lancement des ordres de fabrication ou d'approvisionnement.

Le travail de ce chapitre se base sur les résultats de la thèse (Ould Louly, 2001) et de l'article (Ould Louly et Dolgui, 2008). Nous montrons comment la PSE proposée dans ces travaux peut être adaptée et utilisée dans notre cas. Cette PSE s'appuie sur les calculs des bornes inférieures et supérieures de la fonction objectif et des propriétés de dominance des variables de décision qui nous avons démontré. Nous présentons ensuite quelques résultats expérimentaux afin d'illustrer les performances de la méthode face à différents problèmes.

Nous proposons également une procédure de réduction de l'espace de recherche qui, appliquée avant l'optimisation, permet de réduire le temps de calcul.

## **4.2 Description du problème**

Le système étudié dans ce chapitre est un système d'assemblage à deux niveaux avec un nombre quelconque de composants. Le produit fini est obtenu par l'assemblage des composants de niveau 1, ceux-ci étant eux-mêmes assemblés à partir des composants de niveau 2 (voir la figure 4.1).

Nous limitons notre étude à un seul type de produit fini. La demande est supposée connue ainsi que la date de livraison souhaitée par le client pour cette demande. Nous considérons le cas d'un mono-période. Le coût unitaire de stockage pour chaque type de composant et le coût unitaire de rupture en produit fini sont connus. Les délais d'approvisionnement pour les différents types de composants sont des variables aléatoires discrètes indépendantes. Les lois de distribution sont quelconques, connues et leurs réalisations maximales sont supposées finies.

Nous cherchons donc à obtenir les dates de lancement des ordres au niveau 2 qui minimisent la somme du coût moyen de stockage des composants et du coût moyen de rupture en produit fini, pour le problème PP1, ou la minimisation du coût moyen de stockage des composants sous la contrainte de niveau de service, pour le problème PP2.

Nous utilisons les notations suivantes :

$T$	Date de livraison du produit fini ; sans perte de généralité on pose $T = 0$
$D$	Demande (connue) en produit fini pour la date $T$ ; sans perte de généralité on pose $D = 1$
$N_j$	Nombre de types de composants au niveau $j$ ( $j=1$ ou $2$ )
$c_{i,j}$	Nom du $i^{\text{ème}}$ composant du niveau $j$ ( $j=1$ ou $2$ )
$S_{i,1}$	Ensemble des composants de niveau 2 nécessaires à l'assemblage d'un composant $c_{i,1}$
$n_{i,1}$	Nombre de composants faisant partie de l'ensemble $S_{i,1}$
$L_{i,j}$	Délai d'approvisionnement des composants de type $c_{i,j}$ , (variable aléatoire discrète)
$F_{i,j}(\cdot)$	Fonction de répartition de la variable aléatoire $L_{i,j}$
$u_{i,j}$	Valeur maximale de $L_{i,j}$ , ç.-à.-d. chaque $L_{i,j}$ varie dans un intervalle $[1, u_{i,j}]$
$h_{i,j}$	Coût unitaire de stockage des composants $c_{i,j}$ sur un intervalle de temps unitaire (c.-à-d. pour un composant pour une unité de temps de stockage)
$b$	Coût unitaire de rupture en produit fini sur un intervalle de temps unitaire (c.-à-d. pour un produit fini et pour une unité de temps de retard de livraison)
$x_{i,j}$	Délai d'approvisionnement planifié pour les composants de type $c_{i,j}$ (temps de cycle planifié)
$X_{i,2}$	<b>Variable de décision</b> : Date de lancement du composant de type $c_{i,j}$ au niveau 2
$(x)^+$	$\text{Max}(0, x)$ .

On suppose que l'entreprise lance directement ses commandes aux fournisseurs de niveau 2. Chaque fournisseur de niveau 2 livre les composants nécessaires aux fournisseurs de niveau 1 avec un délai de livraison aléatoire. Les composants de niveau 2 sont alors assemblés pour fabriquer les composants de niveau 1. Les délais d'approvisionnement en composants de niveau 1, qui incluent la fabrication et la livraison, sont également aléatoires. Nous supposons que chaque composant de niveau 2 intervient dans le montage d'un unique composant de niveau 1 (l'hypothèse de la nomenclature arborescente sans composants communs est donc retenue).

Nous supposons aussi que la capacité d'assemblage est infinie. Les délais d'approvisionnement pour un type de composant suivent toujours la même loi quelle que soit la quantité commandée, l'assemblage et la livraison se font en juste à temps.

Nous présentons dans ce qui suit un modèle pour chacun de deux problèmes PP1 et PP2 en exprimant analytiquement les critères à optimiser, et les contraintes à respecter.

### 4.3 Problème PP1

Le problème est donc de trouver la planification optimale pour ce type de systèmes d'assemblage en minimisant le coût engendré par le stockage des produits de niveau 1 et de niveau 2 et par la rupture en produit fini. Dans ce qui suit, nous présentons analytiquement ce coût total moyen à minimiser.

#### 4.3.1 Description du problème

Comme ici, nous considérons le cas d'assemblage au plus tôt : un produit est assemblé dès que tous les composants nécessaires pour son assemblage sont présents, le problème est donc de trouver la date de lancement optimale au niveau 2 afin de satisfaire la demande  $D$  à la date souhaitée  $T$ , en minimisant le coût total moyen (voir la figure 4.2).

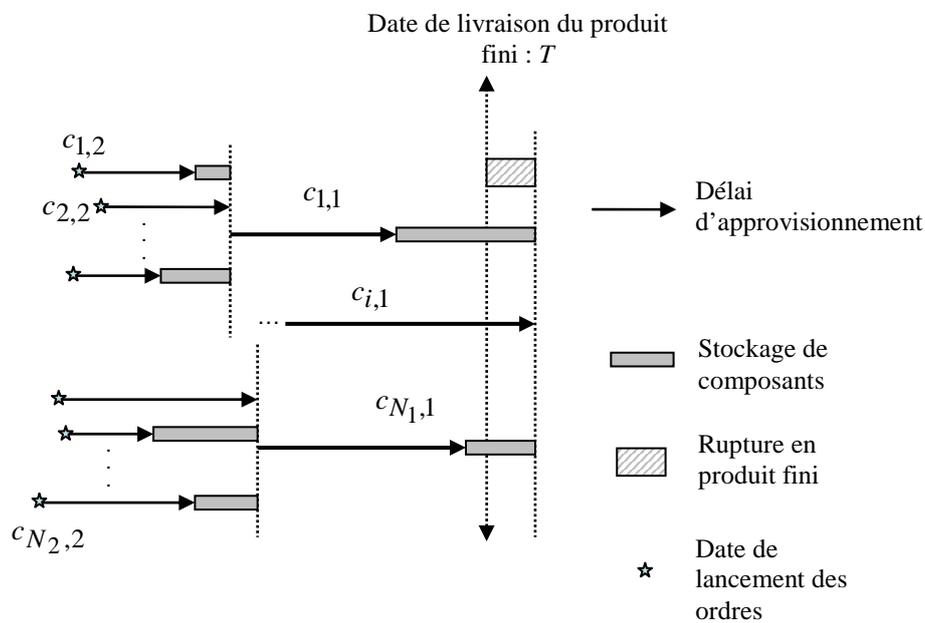


Figure 4.2 : Description du problème PP1

### 4.3.2 Minimisation des coûts

**Proposition 4.1** Pour ce système d'assemblage à deux niveaux, le coût total de PP1 s'exprime comme suit :

$$C(X, L) = H \max_{i=1, \dots, N_1} (M_{i,1} + L_{i,1})^+ - \sum_{i=1}^{N_1} h_{i,1} (M_{i,1} + L_{i,1}) + \sum_{i=1}^{N_1} \left( \sum_{c_{k,2} \in S_{i,1}} h_{k,2} (M_{i,1} - (L_{k,2} - X_{k,2})) \right) \quad (4.1)$$

où,

$$L = (L_{1,1}, \dots, L_{N_1,1}; L_{1,2}, \dots, L_{N_2,2})$$

$$X = (X_{1,2}, \dots, X_{k,2}, \dots, X_{N_2,2})$$

$$M_{i,1} = \max_{c_{k,2} \in S_{i,1}} (L_{k,2} - X_{k,2})$$

$$H = b + \sum_{i=1}^{N_1} h_{i,1}$$

#### Démonstration

Il y a stockage de certaines composants de niveau 2,  $c_{k,2} \in S_{i,1}$ , si tous ces composants n'arrivent pas à la même date, car c'est le dernier composant arrivé qui déclenche l'assemblage. Le coût de stockage des composants  $c_{k,2} \in S_{i,1}$  est donc égal à :

$$\sum_{i=1}^{N_1} \left( \sum_{c_{k,2} \in S_{i,1}} h_{k,2} (M_{i,1} - (L_{k,2} - X_{k,2})) \right) \quad (4.2)$$

où,

$$M_{i,1} = \max_{c_{k,2} \in S_{i,1}} (L_{k,2} - X_{k,2})$$

Il n'y a pas d'autres coûts à considérer que les coûts de stockage puisque aucune date de livraison n'est expressément requise à ce niveau (voir la figure 4.2).

Concernant l'assemblage du produit fini à partir des composants de niveau 1, il peut y avoir à la fois des coûts de stockage de composants et un coût de rupture en produit fini. En effet, les composants de niveau 1 sont stockés au fur et à mesure de leur arrivée jusqu'à ce que tous les composants soient présents (comme au niveau précédent), mais en plus il peut y avoir un coût de rupture si le dernier composant arrive après  $T=0$ , date de livraison demandée du produit fini, c.-à-d. si :  $M_{i,1} + L_{i,1} > 0$ . Le temps de rupture en produit fini se calcule en recherchant le composant pour lequel la valeur

$(M_{i,1} + L_{i,1})^+$  est la plus grande, c.-à-d. :  $\max_{i=1,\dots,N_1} (M_{i,1} + L_{i,1})^+$ . Le coût de rupture en produit fini

vaut donc :

$$b \max_{i=1,\dots,N_1} (M_{i,1} + L_{i,1})^+ \quad (4.3)$$

Par ailleurs, il y a stockage des composants de type  $c_{i,1}$ ,  $i = 1, \dots, N_1$ , pendant le temps s'écoulant entre la date de leur arrivée  $(M_{i,1} + L_{i,1})$  et le maximum entre  $T=0$  et la date à laquelle tous les composants de niveau 1 sont présents :  $\max_{i=1,\dots,N_1} (M_{i,1} + L_{i,1})^+$ . Le coût de stockage des composants de niveau 1 est

donc le suivant :

$$\sum_{i=1}^{N_1} h_{i,1} \left( \max_{i=1,\dots,N_1} (M_i + L_{i,1})^+ - (M_i + L_{i,1}) \right) \quad (4.4)$$

C.Q.F.D.

Les délais d'approvisionnement  $L_{i,j}$  sont des variables aléatoires et donc le coûts ci-dessus (4.1) est une variable aléatoire. Notre objectif est de minimiser l'espérance mathématique de (4.1), c'est-à-dire le coût total moyen  $C(X) = EC(X, L)$ .

**Proposition 4.2** *Le coût total moyen du PP1 s'exprime comme suit :*

$$\begin{aligned} EC(X) = & H \sum_{s \geq 0} \left( 1 - \prod_{i=1}^{N_1} \left( \sum_{o_1+o_2=s} \Pr(L_{i,1} = o_1) \times \prod_{c_{k,2} \in S_{i,1}} F_{k,2}(X_{k,2} + o_2) \right) \right) \\ & + \sum_{i=1}^{N_1} H_i \times \left( \sum_{s \geq 0} \left( 1 - \prod_{c_{k,2} \in S_{i,1}} F_{k,2}(X_{k,2} + s) \right) \right) \\ & - \sum_{i=1}^{N_1} H_i \times \left( \sum_{s \geq 0} \left( 1 - \prod_{c_{k,2} \in S_{i,1}} (1 - F_{k,2}(X_{k,2} - s - 1)) \right) \right) \\ & - \sum_{i=1}^{N_1} \left( \sum_{c_{k,2} \in S_{i,1}} h_{k,2} (E(L_{k,2}) - X_{k,2}) \right) - \sum_{i=1}^{N_1} h_{i,1} E(L_{i,1}) \end{aligned} \quad (4.5)$$

où

$$H = b + \sum_{i=1}^{N_1} h_{i,1}, \quad H_i = \sum_{c_{k,2} \in S_{i,1}} (h_{k,2}) - h_{i,1} \text{ pour } i = 1, \dots, N_1$$

$$X = (X_{1,2}, \dots, X_{i,2}, \dots, X_{N_2})$$

### Démonstration

Pour démontrer la fonction objectif (4.5) nous avons besoin de calculer :

$$E(Z) = E\left(\max_{i=1,\dots,N_1} (M_{i,1} + L_{i,1})^+\right) \text{ et } E(W) = E(M_{i,1})$$

$Z$  est une variable aléatoire discrète positive à nombre fini des valeurs possibles, son espérance mathématique est donc égale à :

$$E(Z) = \sum_{s \geq 0} \Pr(Z > s) = \sum_{s \geq 0} (1 - \Pr(Z \leq s))$$

Donc,

$$E(Z) = \sum_{s \geq 0} \left(1 - \Pr\left(\max_{i=1,\dots,N_1} (Z_i) \leq s\right) \times \Pr(s \geq 0)\right) = \sum_{s \geq 0} \left(1 - \Pr\left(\max_{i=1,\dots,N_1} (Z_i) \leq s\right)\right)$$

Sachant que :

$$\Pr\left(\max_{i=1,\dots,N_1} Z_i \leq s\right) = \Pr(Z_1 \leq s, \dots, Z_i \leq s, \dots, Z_{N_1} \leq s)$$

De plus, pour  $i=1, \dots, N_1$ , les variables aléatoires  $Z_i = (M_i + L_{i,1})$  sont également indépendantes, alors :

$$\begin{aligned} E(Z) &= \sum_{s \geq 0} \left(1 - \prod_{i=1}^{N_1} \Pr(Z_i \leq s)\right) = \sum_{s \geq 0} \left(1 - \prod_{i=1}^{N_1} \Pr(M_{i,1} + L_{i,1} \leq s)\right) \\ &= \sum_{s \geq 0} \left(1 - \prod_{i=1}^{N_1} \sum_{o_1+o_2=s} \Pr(L_{i,1} = o_1) \times \Pr(M_{i,1} \leq o_2)\right) \end{aligned}$$

Rappelons que :  $M_{i,1} = \max_{c_{k,2} \in S_{i,1}} Z_{k,2}$ .

Les variables aléatoires  $Z_{k,2} = L_{k,2} - X_{k,2}$  sont indépendantes, alors :

$$\begin{aligned} E(Z) &= \sum_{s \geq 0} \left(1 - \prod_{i=1}^{N_1} \left\{ \sum_{o_1+o_2=s} \Pr(L_{i,1} = o_1) \times \prod_{c_{k,2} \in S_{i,1}} \Pr(L_{k,2} \leq X_{k,2} + o_2) \right\}\right) \\ &= \sum_{s \geq 0} \left(1 - \prod_{i=1}^{N_1} \left\{ \sum_{o_1+o_2=s} \Pr(L_{i,1} = o_1) \times \prod_{c_{k,2} \in S_{i,1}} F_{k,2}(X_{k,2} + o_2) \right\}\right) \end{aligned}$$

D'autre part,

$$E(W) = E(W^+ - W^-),$$

$W^+$  et  $W^-$  sont des variables aléatoires discrètes positives, donc nous montrons de la manière que  $E(Z)$  :

$$E(W^+) = \sum_{s \geq 0} \left(1 - \Pr\left(\max_{c_{k,2} \in S_{i,1}} (L_{k,2} - X_{k,2}) \leq s\right) \times \Pr(s \geq 0)\right)$$

$$= \sum_{s \geq 0} \left( 1 - \Pr \left( \max_{c_{k,2} \in S_{i,1}} (L_{k,2} - X_{k,2}) \leq s \right) \right)$$

Sachant que :

$$\Pr \left( \max_{c_{k,2} \in S_{i,1}} (w_k) \leq s \right) = \Pr(w_1 \leq s, \dots, w_k \leq s, \dots, w_{N_2} \leq s)$$

De plus, pour  $c_{k,2} \in S_{i,1}$ , les variables aléatoires  $w_k = (L_{k,2} - X_{k,2})$  sont indépendantes, alors :

$$E(W^+) = \sum_{s \geq 0} \left( 1 - \prod_{c_{k,2} \in S_{i,1}} F_{k,2}(X_{k,2} + s) \right)$$

$$E(W^-) = \sum_{s \geq 0} \left( 1 - \Pr \left( \max_{c_{k,2} \in S_{i,1}} (X_{k,2} - L_{k,2}) \leq s \right) \times \Pr(s \geq 0) \right)$$

$$= \sum_{s \geq 0} \left( 1 - \Pr \left( \max_{c_{k,2} \in S_{i,1}} (X_{k,2} - L_{k,2}) \leq s \right) \right)$$

Sachant que :

$$\Pr \left( \max_{c_{k,2} \in S_{i,1}} (w_k) \leq s \right) = \Pr(w_1 \leq s, \dots, w_k \leq s, \dots, w_{N_2} \leq s)$$

De plus, pour  $c_{k,2} \in S_{i,1}$ , les variables aléatoires  $w_k = (L_{k,2} - X_{k,2})$  sont indépendantes, alors :

$$E(W^-) = \sum_{s \geq 0} \left( 1 - \prod_{c_{k,2} \in S_{i,1}} \Pr((X_{k,2} - L_{k,2}) \leq s) \right)$$

$$E(W^-) = \sum_{s \geq 0} \left( 1 - \prod_{c_{k,2} \in S_{i,1}} (1 - \Pr(L_{k,2} \leq X_{k,2} - s - 1)) \right)$$

$$E(W^-) = \sum_{s \geq 0} \left( 1 - \prod_{c_{k,2} \in S_{i,1}} (1 - F_{k,2}(X_{k,2} - s - 1)) \right)$$

C.Q.F.D.

### 4.3.3 Optimisation

#### 4.3.3.1 Problèmes à résoudre

Notre problème peut donc se modéliser par le problème d'optimisation suivant :

**Problème PP1 :**

$$\text{Min } EC(X)$$

Sous les contraintes :

$$X_{k,2} = x_{k,2} + x_{i,1} \quad (4.6)$$

$$U_{k,2} = u_{k,2} + u_{i,1} \quad (4.7)$$

$$2 \leq X_{i,2} \leq U_{i,2} \quad (4.8)$$

$$1 \leq x_{i,j} \leq u_{i,j} \quad (4.9)$$

Avec :  $j=1,2$ ,  $i=1,2,\dots, N_j$ , et  $c_{k,2} \in S_{i,1}$

Nous nous trouvons face à une fonction non linéaire à variables entières. Dès que l'énumération explicite de tout l'espace de recherche devient impossible (c.-à-d. quand la taille du problème est importante), il est nécessaire d'utiliser des outils de la recherche opérationnelle performants afin de pouvoir trouver soit la solution optimale, soit au moins une bonne solution obtenue à l'aide d'une méthode approchée.

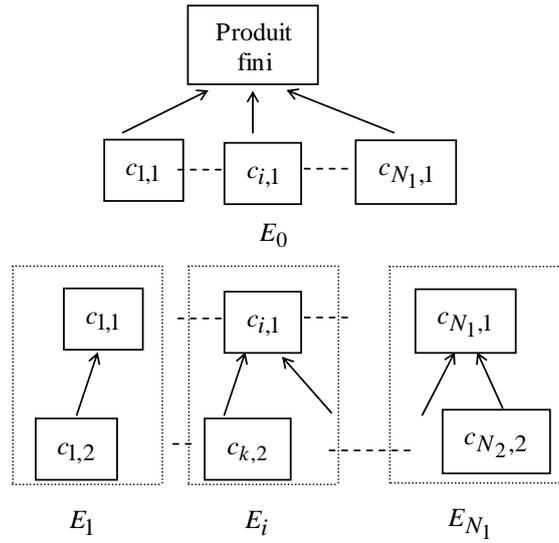
Nous présentons par la suite une procédure d'optimisation exacte et efficace de type par Séparation et Évaluation (PSE) que nous avons développé pour résoudre ce problème.

Nous commençons dans la section suivante par présenter une procédure de décomposition du problème qui permet de réduire l'espace de recherche de solution avant de commencer l'optimisation.

#### **4.3.3.2 Pré-traitement : procédure de réduction de l'espace de recherche**

Nous proposons une procédure de réduction de l'espace de recherche qui sera utilisée avant de commencer l'optimisation. Cette procédure permet de réduire les intervalles des valeurs possibles pour  $X_{i,2}$  qui sont initialement définis comme suit :  $2 \leq X_{i,2} \leq U_{i,2}$ ,  $i=1,2,\dots, N_2$ .

Afin d'obtenir des limites inférieures et supérieures des variables de décision  $X_{i,2}$ , nous proposons de décomposer le problème niveau par niveau en sous-problèmes d'assemblage à un seul niveau notés  $E_i$ , ( $i=0,\dots, N_1$ ) comme illustré dans la figure 4.3.



**Figure 4.3 :** Décomposition du problème PP1

Le problème d'assemblage à un seul niveau avec coût de rupture a été étudié dans Ould Louly et Dolgui (2008). Une PSE a été proposée, nous la présentons plus loin dans ce texte. Ici, mentionnons simplement que dans la première étape, nous pouvons résoudre le problème  $E_0$  en utilisant cette PSE.

Soit :  $X^* = (x_{1,1}^*, \dots, x_{i,1}^*, \dots, x_{N_1,1}^*)$  la solution optimale de  $E_0$ .

Cette première étape permet de réduire l'espace de recherche des solutions : initialement les variables de décision sont dans les intervalles :  $2 \leq X_{i,2} \leq U_{i,2}$ ,  $i=1,2,\dots, N_2$ , elles sont maintenant dans les intervalles  $x_{i,1}^* \leq X_{k,2} \leq U_{k,2}$ ,  $i=1,2,\dots, N_1$  et  $c_{k,2} \in S_{i,1}$ .

En seconde étape, nous résolvons chaque problème  $E_i$  en utilisant la même PSE. Comme délai de livraison pour  $E_i$  nous utilisons la valeur  $x_{i,1}^*$  obtenue par  $E_0$ , et ce pour tout  $i=1,\dots, N_1$ . Pour  $E_0$  le coût de retard est  $b$ , mais pour  $E_i$ , il faut calculer le coût de retard  $b_{i,1}$ .

**Proposition 4.3** Lorsque le retard au niveau 2 est égal à  $y$  ( $y=1,\dots,u$ ) unités de temps, correspondant au délai de livraison du composant  $c_{i,1}$ , alors le coût de retard du problème  $E_i$  est :

$$b_{i,1}(y) = -yh_{i,1} + H \sum_{k \geq 0} \left( \sum_{s=0}^{y-1} \Pr(L_{i,1} = x_{i,1}^* + k - s) \times \prod_{\substack{j=1 \\ j \neq i}}^{N_1} F_{j,1}(x_{j,1}^* + k) \right) \quad (4.10)$$

où,

$$H = b + \sum_{i=1}^{N_1} h_{i,1}$$

$$u = \max_{c_{k,2} \in \mathcal{S}_{i,1}} (u_{k,2} - 1)$$

$(x_{1,1}^*, \dots, x_{i,1}^*, \dots, x_{N_1,1}^*)$  est la solution optimale de  $E_0$ .

#### Démonstration

Soit  $EC_0(x_{1,1}^*, \dots, x_{i,1}^*, \dots, x_{N_1,1}^*)$  le coût de la solution optimal de  $E_0$ .

Donc :

$$b_{i,1}(y) = EC_0(x_{1,1}^*, \dots, x_{i,1}^* - y, \dots, x_{N_1,1}^*) - EC_0(x_{1,1}^*, \dots, x_{i,1}^*, \dots, x_{N_1,1}^*) \quad (4.11)$$

D'après Ould Louly et Dolgui (2008):

$$EC_0(x_{1,1}^*, \dots, x_{i,1}^*, \dots, x_{N_1,1}^*) = \sum_{i=1}^{N_1} h_{i,1} (x_{i,1}^* - E(L_{i,1})) + H \sum_{k \geq 0} \left( 1 - \prod_{i=1}^{N_1} F_{i,1}(x_{i,1}^* + k) \right) \quad (4.12)$$

où :

$$H = b + \sum_{i=1}^{N_1} h_{i,1} .$$

Finalement, en utilisant (4.12) dans (4.11), nous obtenons (4.10).

C.Q.F.D.

**Proposition 4.4** Les propriétés des coûts de retard  $b_{i,1}(y)$ ,  $\forall y = 1, \dots, u$  sont :

$$0 \leq b_{i,1}(y) \leq -y \times h_{i,1} + y \times H \quad (4.13)$$

$$b_{i,1}(1) \leq \dots \leq b_{i,1}(y) \leq \dots \leq b_{i,1}(u) \quad (4.14)$$

### Démonstration

En utilisant (4.10), nous obtenons :

$$b_{i,1}(y) \leq -y \times h_{i,1} + y \times H$$

Considérant (4.11), c.-à.-d. que la condition nécessaire d'optimalité de  $(x_{1,1}^*, \dots, x_{i,1}^*, \dots, x_{N_1,1}^*)$  est :  $0 \leq b_{i,1}(y)$ . Donc, nous démontrons la propriété (4.13).

Pour démontrer (4.14), i.e. la fonction  $b_{i,1}(y)$  est une fonction croissante par rapport à  $y$ , nous utiliserons la fonction  $G_i^-(X)$  qui a été introduite dans Ould Louly et Dolgui (2008) comme suit :

$$G_i^-(X) = EC_0(x_{1,1}^*, \dots, x_{i,1}^* - 1, \dots, x_{N_1,1}^*) - EC_0(x_{1,1}^*, \dots, x_{i,1}^*, \dots, x_{N_1,1}^*)$$

En utilisant cette fonction,  $b_{i,1}(y)$  peut être réécrite comme suit :

$$b_{i,1}(y) = \sum_{s=0}^{y-1} G_i^-(x_{1,1}^*, \dots, x_{i,1}^* - s, \dots, x_{N_1,1}^*)$$

où,

$$G_i^-(X) = -h_{i,1} + H \sum_{k \geq 0} \Pr(L_{i,1} = x_{i,1} + k) \prod_{\substack{j=1 \\ j \neq i}}^{N_1} F_{j,1}(x_{j,1} + k) \geq 0$$

Pour montrer (4.14), nous avons besoin de montrer que :

$$b_{i,1}(y) - b_{i,1}(y-1) \geq 0$$

$$\text{Comme : } b_{i,1}(y) - b_{i,1}(y-1) = G_i^-(x_{1,1}^*, \dots, x_{i,1}^* - (y-1), \dots, x_{N_1,1}^*)$$

D'après Ould Louly et Dolgui (2008),  $G_i^-(X)$  est une fonction décroissante de  $x_{i,1}$ .

$$G_i^-(x_{1,1}^*, \dots, x_{i,1}^* - (y-1), \dots, x_{N_1,1}^*) \geq G_i^-(x_{1,1}^*, \dots, x_{i,1}^*, \dots, x_{N_1,1}^*)$$

Finalement, en tenant compte de la définition de  $G_i^-(X)$  et de la condition nécessaire d'optimalité de

$$(x_{1,1}^*, \dots, x_{i,1}^*, \dots, x_{N_1,1}^*), \text{ nous obtenons :}$$

$$G_i^-(x_{1,1}^*, \dots, x_{i,1}^*, \dots, x_{N_1,1}^*) \geq 0.$$

Donc :  $b_{i,1}(y) - b_{i,1}(y-1) \geq 0$

C.Q.F.D.

**Proposition 4.5** Le coût total moyen pour chaque  $E_i$  ( $i=1,\dots,N_1$ ) considéré séparément est le suivant :

$$\begin{aligned}
 EC_i(X_i) = & \sum_{c_{k,2} \in S_{i,1}} h_{k,2} \times \left( \sum_{s \geq 0} \left( 1 - \prod_{c_{k,2} \in S_{i,1}} F_{k,2}(x_{k,2} + s) \right) - (E(L_{k,2}) - x_{k,2}) \right) \\
 & + \sum_{y=1}^u b_{i,1}(y) \times \left( \prod_{c_{k,2} \in S_{i,1}} F_{k,2}(x_{k,2} + y) - \prod_{c_{k,2} \in S_{i,1}} F_{k,2}(x_{k,2} + y - 1) \right)
 \end{aligned} \tag{4.15}$$

où,

$X_i$  est le vecteur des valeurs  $x_{k,2}$  pour tout  $c_{k,2} \in S_{i,1}$  et ( $i=1,\dots,N_1$ ).

### Démonstration

Le coût pour chaque problème  $E_i$  considéré séparément est égal à la somme des coûts de stockage des composants :

$$\sum_{c_{k,2} \in S_{i,1}} h_{k,2} \times \left( m_{i,1}^+ - (L_{k,2} - x_{k,2}) \right)$$

où :

$$m_{i,1} = \max_{c_{k,2} \in S_{i,1}} (L_{k,2} - x_{k,2})$$

plus le coût de retard, si l'opération d'assemblage a été effectuée après la date prévue ( $-x_{i,1}^*$ ). Ce coût de retard est égal à :

$$\sum_{y=1}^u b_{i,1}(y) \times \Pr(m_{i,1}^+ = y)$$

Le coût total pour  $E_i$ ,  $i=1,\dots,N_1$  est donc :

$$C_i(X_i, L_i) = \sum_{c_{k,2} \in S_{i,1}} h_{k,2} \times \left( m_{i,1}^+ - (L_{k,2} - x_{k,2}) \right) + \sum_{y=1}^u b_{i,1}(y) \times \Pr(m_{i,1}^+ = y)$$

où :

$X_i$  est le vecteur des valeurs  $x_{k,2}$  pour tout  $c_{k,2} \in S_{i,1}$

$L_i$  est le vecteur des valeurs  $L_{k,2}$  pour tout  $c_{k,2} \in S_{i,1}$

Les coûts  $C_i(X_i, L_i)$  sont des variables aléatoires. Nous utilisons leurs espérances mathématiques  $EC_i(X_i, L_i)$  qui sont calculées d'une manière similaire à la démonstration de la proposition 4.2.

C.Q.F.D.

Soit  $[A, B]$  l'espace des solutions réalisables des problèmes PP1 et PP2, où  $A$  est le vecteur  $(a_1, a_2, \dots, a_{N_2})$  des valeurs minimales possibles et  $B$  est le vecteur  $(b_1, b_2, \dots, b_{N_2})$  des valeurs maximales possibles des variables de décision, i.e.  $a_i \leq X_{i,2} \leq b_i, i=1, 2, \dots, N_2$ .

Par définition :

$$2 \leq X_{i,2} \leq U_{i,2}, i=1, 2, \dots, N_2.$$

Notre objectif est de réduire cette espace  $[A, B]$  le plus possible avant de commencer l'optimisation.

Considérant que chaque type de composants de niveau 2 est utilisé pour assembler un seul type de composants de niveau 1, l'espace des solutions réalisables de niveau 2 peut être obtenu en résolvant les sous-ensembles  $E_i, i=1, 2, \dots, N_2$ , séparément. Cet espace peut être représenté comme suit :  $[LL_2, UL_2]$ , avec:

$$LL_2 = (ll_{1,2}, \dots, ll_{i,2}, \dots, ll_{N_2,2})$$

$$UL_2 = (ul_{1,2}, \dots, ul_{i,2}, \dots, ul_{N_2,2})$$

$ll_{i,2}$  et  $ul_{i,2}$  sont les limites inférieurs et supérieurs des variables  $x_{i,2}$ , pour  $i=1, \dots, N_2$ .

**Proposition 4.6** Si  $c_{k,2} \in S_{i,1}$ , les limites inférieurs et supérieurs des  $x_{k,2}$  peut être obtenue comme suit :

- la solution optimale de  $E_i$  avec le coût de retard  $b=0$  donne la limite inférieure  $ll_{k,2}$  de  $x_{k,2}$ , pour  $k=1, \dots, N_2$ .
- la solution optimale de  $E_i$  avec le coût de retard  $b=H - h_{i,1}$  donne la limite supérieure  $ul_{k,2}$  de  $x_{k,2}$ , pour  $k=1, \dots, N_2$ .

### Démonstration

La démonstration de la proposition 4.6 est basée sur la déclaration suivante :

Considérons un système d'assemblage  $E$  à un seul niveau avec  $n$  types de composants, les coûts de stockage unitaire  $h_k, k=1, 2, \dots, n$ , et le coût de rupture unitaire  $b$ . Si le coût de stockage  $b$  pour ce

système augmente, alors les valeurs optimales des variables de décision  $x_k$  augmentent aussi. Nous pouvons démontrer cette déclaration comme suit.

Soit  $b$  et  $b'$  les coûts de rupture unitaires, tel que :  $b \leq b'$  ;

$x_k$  et  $x'_k$  les solutions optimales de  $E$  avec le coût de rupture  $b$  et  $b'$  , respectivement.

Nous utiliserons la démonstration par l'absurde en supposant que :

$$x'_k \leq x_k, k=1,2,\dots,n \quad (4.16)$$

Les expressions suivantes (4.17) et (4.18) sont des conditions nécessaires d'optimalité :

$$EC(\{x'_1, \dots, x'_k, \dots, x'_n\}, b) - EC(\{x_1, \dots, x_k, \dots, x_n\}, b) \geq 0 \quad (4.17)$$

$$EC(\{x_1, \dots, x_k, \dots, x_n\}, b') - EC(\{x'_1, \dots, x'_k, \dots, x'_n\}, b') \geq 0 \quad (4.18)$$

Utilisant (4.12), l'inégalité (4.17) peut être réécrite comme suit :

$$\sum_{i=1}^n h_i (x'_i - x_i) + H \sum_{k \geq 0} \left( \prod_{i=1}^n F_i(x_i + k) - \prod_{i=1}^n F_i(x'_i + k) \right) \geq 0 \quad (4.19)$$

$$\text{où : } H = b + \sum_{i=1}^n h_i$$

De la même manière, l'inégalité (4.18) peut être réécrite comme suit :

$$\sum_{i=1}^n h_i (x_i - x'_i) - H' \sum_{k \geq 0} \left( \prod_{i=1}^n F_i(x_i + k) - \prod_{i=1}^n F_i(x'_i + k) \right) \geq 0 \quad (4.20)$$

$$\text{où : } H' = b' + \sum_{i=1}^n h_i$$

Utilisant (4.17) dans (4.16), nous obtenons :

$$\left( 1 - \frac{H}{H'} \right) \times \sum_{i=1}^n h_i (x'_i - x_i) \geq 0$$

Etant donné que  $H' \geq H$  (car  $b \leq b'$ ), nous obtenons :  $\sum_{i=1}^n h_i (x'_i - x_i) \geq 0$

Ce qui est en contradiction avec (4.16).

D'après la propriété (4.17) de la proposition 4.6 :  $\forall y = 1, \dots, u$  et  $i = 1, 2, \dots, n$ ,

$$0 \leq b_{i,1}(y) \leq y \times H - y \times h_{i,1} \quad (4.21)$$

Les limites inférieures et supérieures des variables  $x_{k,2}$  peuvent donc être obtenues en résolvant les problèmes  $E_i$  avec  $b=0$  et  $b=H-h_{i,1}$ .

C.Q.F.D.

La proposition 4.6 permet de calculer les limites inférieures et supérieures des dates de lancement des ordres au niveau 2. Lorsque les limites inférieures et supérieures des variables  $x_{k,2}$ ,  $k=1,\dots,N_2$ , sont connues, nous pouvons calculer les limites inférieures et supérieures des variables de décision  $X_{k,2}$  comme suit :

$$a_k = x_{i,1}^* + ll_{k,2} \quad (4.22)$$

$$b_k = x_{i,1}^* + ul_{k,2} \quad (4.23)$$

pour  $k=1,2,\dots,N_2$ ;  $i=1,2,\dots,N_1$ , et  $c_{k,2} \in S_{i,1}$

Nous avons présenté une procédure de réduction de l'espace de recherche initial. Nous cherchons maintenant des propriétés de dominance, dans le but de réduire l'espace de recherche de solutions à chaque étape de la procédure de la PSE, et nous calculons deux bornes inférieures, et une borne supérieure.

Avant de détailler les relations montrant les propriétés de dominance, nous introduisons la définition des accroissements partiels :

#### 4.3.3.3 Accroissements partiels

Nous utilisons ici la même approche que dans Ould Louly et Dolgui (2008), mais nous généralisons pour les systèmes à deux niveaux. Les accroissements partiels à droite et à gauche par rapport à la variable  $X_{k,2}$  sont définis comme suit :

$$G_k^+(X) = EC(X_{1,2}, \dots, X_{k,2} + 1, \dots, X_{N_2,2}) - EC(X_{1,2}, \dots, X_{k,2}, \dots, X_{N_2,2}), \quad (4.24)$$

$$G_k^-(X) = EC(X_{1,2}, \dots, X_{k,2} - 1, \dots, X_{N_2,2}) - EC(X_{1,2}, \dots, X_{k,2}, \dots, X_{N_2,2}). \quad (4.25)$$

Notons que la fonction  $G_k^-(X)$  n'est définie que pour les vecteurs  $X$  pour lesquels la  $k^e$  composante  $X_{k,2}$  est strictement supérieure à 1. De la même manière,  $G_k^+(X)$  n'est définie que pour les vecteurs  $X$  pour lesquels la  $k^e$  composante  $X_{k,2}$  est strictement inférieure à  $U_{k,2}$ .

**Proposition 4.7** Les accroissements partiels  $G_k^+(X)$  et  $G_k^-(X)$  sont des fonctions qui vérifient les propriétés suivantes :

- (i)  $G_k^+(X)$  est une fonction croissante par rapport à  $X_{k,2}$  et décroissante par rapport à  $X_{j,2}$  pour  $j \neq k$
- (ii)  $G_k^-(X)$  est une fonction décroissante par rapport à  $X_{k,2}$  et croissante par rapport à  $X_{j,2}$  pour  $j \neq k$

### Démonstration

Pour  $c_{k,2} \in S_{i,1}$ ,

$$\begin{aligned}
G_k^+(X) = & h_{k,2} - H \sum_{s \geq 0} \left( \prod_{\substack{z=1 \\ z \neq i}}^{N_1} \left( \sum_{o_1+o_2=s} \Pr(L_{z,1} = o_1) \times \prod_{c_{r,2} \in P_{z,1}} F_{r,2}(X_{r,2} + o_2) \times \right. \right. \\
& \left. \left. \sum_{o_1+o_2=s} \Pr(L_{i,1} = o_1) \times \prod_{\substack{c_{j,2} \in P_{i,1} \\ j \neq k}} F_{j,2}(X_{j,2} + o_2) \times \Pr(L_{k,2} = X_{k,2} + o_2 + 1) \right) \right) \\
& - H_{i,1} \times \sum_{s \geq 0} \left( \Pr(L_{k,2} = X_{k,2} + s + 1) \times \prod_{\substack{c_{j,2} \in P_{i,1} \\ j \neq k}} F_{j,2}(X_{j,2} + s) \right) \\
& - H_{i,1} \times \sum_{s \geq 0} \left( \Pr(L_{k,2} = X_{k,2} - s) \times \prod_{\substack{c_{j,2} \in P_{i,1} \\ j \neq k}} (1 - F_{j,2}(X_{j,2} - s - 1)) \right) \quad (4.26)
\end{aligned}$$

L'expression (4.26) montre que la fonction  $G_k^+(X)$  est décroissante par rapport à  $X_{j,2}$  pour  $j$  différent de  $k$ .

Pour démontrer que  $G_k^+(X)$  est croissante par rapport à  $X_{k,2}$ , il suffit de montrer que la fonction

$R_{k,2}^+(X_{1,2}, \dots, X_{k,2}, \dots, X_{N_2,2})$  définie ci-après est toujours positive.

$$R_{k,2}^+(X_{1,2}, \dots, X_{k,2}, \dots, X_{N_2,2}) = G_k^+(X_{1,2}, \dots, X_{k,2} + 1, \dots, X_{N_2,2}) - G_k^+(X_{1,2}, \dots, X_{k,2}, \dots, X_{N_2,2}) \quad (4.27)$$

En développant  $R_{k,2}^+(X_{1,2}, \dots, X_{k,2}, \dots, X_{N_2,2})$ , nous obtenons la relation suivante :

$$R_{k,2}^+ = \sum_{l=1}^3 R_l$$

Nous détaillerons ci-après tous les termes de la somme de cette expression.

$$\begin{aligned}
R_1 = & H \prod_{\substack{z=1 \\ z \neq i}}^{N_1} \left( \sum_{o_1+o_2=0} \Pr(L_{z,1} = o_1) \times \prod_{c_{r,2} \in P_{z,1}} F_{r,2}(X_{r,2} + o_2) \times \sum_{o_1+o_2=0} \Pr(L_{i,1} = o_1) \times \prod_{\substack{c_{j,2} \in P_{i,1} \\ j \neq k}} F_{j,2}(X_{j,2} + o_2) \right. \\
& \times \sum_{o_1+o_2=0} \Pr(L_{i,1} = o_1) \times \prod_{\substack{c_{j,2} \in P_{i,1} \\ j \neq k}} F_{j,2}(X_{j,2} + o_2) \times \Pr(L_{k,2} = X_{k,2} + o_2 + 1) \left. \right) \\
& + H \sum_{s \geq 0} \left( \prod_{\substack{z=1 \\ z \neq i}}^{N_1} \left( \sum_{o_1+o_2=s} \Pr(L_{z,1} = o_1) \times \prod_{c_{r,2} \in P_{z,1}} F_{r,2}(X_{r,2} + 1 + o_2) \right. \right. \\
& \times \sum_{o_1+o_2=s} \Pr(L_{i,1} = o_1) \times \prod_{\substack{c_{j,2} \in P_{i,1} \\ j \neq k}} F_{j,2}(X_{j,2} + o_2 + 1) \times \Pr(L_{k,2} = X_{k,2} + o_2 + 2) \left. \right) \\
& - H \sum_{s \geq 0} \left( \prod_{\substack{z=1 \\ z \neq i}}^{N_1} \left( \sum_{o_1+o_2=s} \Pr(L_{z,1} = o_1) \times \prod_{c_{r,2} \in P_{z,1}} F_{r,2}(X_{r,2} + o_2) \right. \right. \\
& \times \sum_{o_1+o_2=s} \Pr(L_{i,1} = o_1) \times \prod_{\substack{c_{j,2} \in P_{i,1} \\ j \neq k}} F_{j,2}(X_{j,2} + o_2) \times \Pr(L_{k,2} = X_{k,2} + o_2 + 2) \left. \right) \left. \right)
\end{aligned}$$

$$\begin{aligned}
R_2 = & H_{i,1} \times \Pr(L_{k,2} = X_{k,2} + 1) \prod_{\substack{c_{j,2} \in P_{i,1} \\ j \neq k}} F_{j,2}(X_{j,2}) \\
& + H_{i,1} \times \sum_{s \geq 0} \left( \Pr(L_{k,2} = X_{k,2} + s + 2) \times \left( \prod_{\substack{c_{j,2} \in P_{i,1} \\ j \neq k}} F_{j,2}(X_{j,2} + s + 1) - \prod_{\substack{c_{j,2} \in P_{i,1} \\ j \neq k}} F_{j,2}(X_{j,2} + s) \right) \right) \\
R_3 = & H_{i,1} \times \Pr(L_{k,2} = X_{k,2}) \times \left( 1 - \prod_{\substack{c_{j,2} \in P_{i,1} \\ j \neq k}} F_{j,2}(X_{j,2} - 1) \right)
\end{aligned}$$

$$+ H_{i,1} \times \sum_{s \geq 0} \left( \Pr(L_{k,2} = X_{k,2} + 1 - s) \times \left( \prod_{\substack{c_{j,2} \in P_{i,1} \\ j \neq k}} (1 - F_{j,2}(X_{j,2} - s - 2)) - \prod_{\substack{c_{j,2} \in P_{i,1} \\ j \neq k}} (1 - F_{j,2}(X_{j,2} - s - 1)) \right) \right)$$

Etant donné que pour tout  $v \in Z$  :

$$\prod_k F_{k,2}(X_{k,2} + v + 1) - \prod_k F_{k,2}(X_{k,2} + v) \geq 0,$$

$R_1, R_2$  et  $R_3$  sont positives et par conséquent  $R_{k,2}^+(X)$  est positive et  $G_k^+(X)$  est une fonction croissante de  $X_{k,2}$ .

Les relations (4.24) et (4.25) nous permettent d'établir la relation suivante :

$$G_k^-(X) = -G_k^+(X_{1,2}, \dots, X_{k,2} - 1, \dots, X_{N_2,2}) \quad (4.28)$$

La propriété (ii) peut être alors déduite de la propriété (i).

C.Q.F.D.

#### 4.3.3.4 Propriétés de dominance

D'après la définition de  $G_k^+(X)$  et  $G_k^-(X)$ , nous pouvons dire que les expressions suivantes sont des conditions nécessaires pour que le vecteur  $X$  donne le minimum de la fonction coût  $EC(X)$  :

$$G_k^+(X) \geq 0, k=1, \dots, N_2 \quad (4.29)$$

$$G_k^-(X) \geq 0, k=1, \dots, N_2 \quad (4.30)$$

Ces conditions ne sont pas suffisantes, plusieurs minima locaux peuvent les vérifier.

La proposition suivante définit les propriétés de dominance que nous utilisons pour le problème PP1.

**Proposition 4.8** Les propriétés de dominance sont les suivantes :

- (i) si  $\exists k$  tel que  $G_k^+(A) < 0$ , alors chaque solution  $X$  de  $[A, B]$  avec  $X_{k,2} = a_k$  est dominée.
- (ii) si  $\exists k$  tel que  $G_k^-(B) < 0$ , alors chaque solution  $X$  de  $[A, B]$  avec  $X_{k,2} = b_k$  est dominée.

### Démonstration

En effet, si  $\exists k$  tel que  $G_k^+(A) < 0$ , alors le vecteur  $A$  est dominé par le vecteur  $(a_1, \dots, a_k + 1, \dots, a_{N_2})$  car la définition de la fonction  $G_k^+(X)$  donnera l'inégalité stricte suivante :

$EC(a_1, \dots, a_k + 1, \dots, a_{N_2}) < EC(a_1, \dots, a_k, \dots, a_{N_2})$ . En plus, pour tout vecteur  $X$  de  $[A, B]$  tel que  $X_{k,2} = a_k$ , nous aurons l'inégalité suivante :

$$G_k^+(X_{1,2}, \dots, X_{k-1,2}, a_k, X_{k+1,2}, \dots, X_{N_2,2}) \leq G_k^+(a_1, \dots, a_k, \dots, a_{N_2}) < 0,$$

car la fonction  $G_k^+(X)$  est décroissante par rapport à  $X_{j,2}$  pour  $j$  différent de  $k$ . Le vecteur  $X$  de  $[A, B]$  tel que  $X_{k,2} = a_k$  est donc, à son tour, dominé par le vecteur  $(X_{1,2}, \dots, X_{k-1,2}, a_k + 1, X_{k+1,2}, \dots, X_{N_2,2})$ . La propriété (i) est donc vérifiée, la propriété (ii) se démontre de la même façon.

C.Q.F.D.

Il résulte de ces propriétés que si  $\exists k$  tel que  $G_k^+(A) < 0$  et  $a_k < b_k$ , alors l'ensemble des solutions  $[A, B]$  peut être remplacé par son sous ensemble où  $a_k$  est remplacé par  $a_k + 1$  et si  $\exists k$  tel que  $G_k^+(B) < 0$ , et  $a_k < b_k$ , alors l'ensemble des solutions  $[A, B]$  peut être remplacé par son sous ensemble où  $b_k$  est remplacé par  $b_k - 1$ .

Ces propriétés de dominance peuvent être utilisées pour développer des procédures efficaces de coupe pour la PSE. En effet, après la division d'un noeud, dans une PSE, deux noeuds (descendants) sont créés. Pour chaque noeud, les propriétés de dominance sont utilisées pour réduire les espaces de recherche correspondant avant de passer au prochain branchement.

Les descendants du noeud  $[A, B]$  sont obtenues en partageant l'ensemble des solutions correspondants en deux sous-ensembles  $[A, B^1]$  et  $[A^1, B]$  respectivement  $a_k \leq X_{k,2} \leq \frac{a_k + b_k}{2}$  et

$$\frac{a_k + b_k}{2} + 1 \leq X_{k,2} \leq b_k.$$

### Procédure de coupe inférieure

#### **Début**

réduit  $\leftarrow$  faux

dominé  $\leftarrow$  faux

**Tant que** (non réduit et non dominé)

réduit  $\leftarrow$  vrai

**Pour**  $k= 1$  à  $N_2$  faire  
     **Tant que** ( $G_k^+(A) < 0$  et dominé = faux) **faire**  
         réduit  $\leftarrow$  faux  
         si ( $a_k = b_k$ ) dominé  $\leftarrow$  vrai  
         sinon  $a_k \leftarrow a_k + 1$   
     **Fin Tant que**  
**Fin Pour**  
**Fin Tant que**  
**Fin**

*Procédure de coupe supérieure*

**Début**  
     réduit  $\leftarrow$  faux  
     dominé  $\leftarrow$  faux  
     **Tant que** (non réduit et non dominé)  
         réduit  $\leftarrow$  vrai  
         **Pour**  $k = 1$  à  $N_2$  **faire**  
             **Tant que** ( $G_k^-(B) < 0$  et dominé = faux) **faire**  
                 réduit  $\leftarrow$  faux  
                 si ( $a_k = b_k$ ) dominé  $\leftarrow$  vrai  
                 sinon  $b_k \leftarrow b_k - 1$   
             **Fin Tant que**  
         **Fin Pour**  
     **Fin Tant que**  
**Fin**

**4.3.3.5 Bornes inférieures**

Nous allons maintenant présenter deux bornes inférieures de la fonction objectif que nous utiliserons dans nos algorithmes d'optimisation. Ces bornes s'appuient sur l'étude des accroissements partiels des fonctions objectifs. Cette fois ci c'est encore nous procédons par analogie avec (Ould Louly, 2001) voire également (Ould Louly et Dolgui, 2008).

**Proposition 4.9** Les valeurs  $BI_1$  et  $BI_2$  sont deux bornes inférieures pour la fonction coût sur l'ensemble  $[A, B]$  :

$$BI_1 = EC(A) + \sum_{k=1}^{N_2} \left( (b_k - a_k) \times \min \left( G_k^+ (b_1, \dots, b_{k-1}, a_k, \dots, a_{N_2}), 0 \right) \right) \quad (4.31)$$

$$BI_2 = C(B) + \sum_{k=1}^{N_2} \left( (b_k - a_k) \times \min \left( G_k^- (a_1, \dots, a_{k-1}, b_k, \dots, b_{N_2}), 0 \right) \right) \quad (4.32)$$

### Démonstration

Nous allons démontrer que les valeurs  $BI_1$  et  $BI_2$  minorent la fonction objectif sur l'ensemble  $[A, B]$ .

Pour chaque vecteur  $X \in [A, B]$ , nous obtenons:

$$EC(X) - EC(A) = \sum_{k=1}^{N_2} \left( \sum_{s=0}^{X_{k,2} - a_k - 1} G_k^+ (X_{1,2}, \dots, X_{k-1,2}, a_k + s, \dots, a_{N_2}) \right)$$

En utilisant le fait que  $G_k^+$  est croissante par rapport à  $X_{k,2}$  et décroissante par rapport à  $X_{j,2}$  pour  $j \neq k$ , on a :

$$G_k^+ (X_{1,2}, \dots, X_{k-1,2}, a_k + s, \dots, a_{N_2}) \geq G_k^+ (b_1, \dots, b_{k-1}, a_k, \dots, a_{N_2})$$

Donc, nous obtenons :

$$\begin{aligned} EC(X) - EC(A) &\geq \sum_{k=1}^{N_2} \left( (X_{k,2} - a_k) G_k^+ (b_1, \dots, b_{k-1}, a_k, \dots, a_{N_2}) \right) \\ &\geq \sum_{k=1}^{N_2} \left( (b_k - a_k) \times \min \left( G_k^+ (b_1, \dots, b_{k-1}, a_k, \dots, a_{N_2}), 0 \right) \right) \end{aligned}$$

$$\text{Donc : } BI_1 = EC(A) + \sum_{k=1}^{N_2} \left( (b_k - a_k) \times \min \left( G_k^+ (b_1, \dots, b_{k-1}, a_k, \dots, a_{N_2}), 0 \right) \right)$$

De la même manière pour chaque vecteur  $X \in [A, B]$ , nous obtenons :

$$\sum_{k=1}^{N_2} \left( \sum_{s=1}^{b_k - X_{k,2}} G_k^- (X_{1,2}, \dots, X_{k-1,2}, X_{k,2} + s, b_{k+1}, \dots, b_{N_2}) \right)$$

En utilisant le fait que  $G_k^-$  est décroissante par rapport à  $X_{k,2}$  et croissante par rapport à  $X_{j,2}$  pour  $j \neq k$ , on a :

$$\begin{aligned}
EC(X) - EC(B) &\geq \sum_{k=1}^{N_2} \left( (b_k - X_{k,2}) G_k^-(a_1, \dots, a_{k-1}, b_k, \dots, b_{N_2}) \right) \\
&\geq \sum_{k=1}^{N_2} \left( (b_k - a_k) \times \min \left( G_k^-(a_1, \dots, a_{k-1}, b_k, \dots, b_{N_2}), 0 \right) \right) \\
\text{Donc : } BI_2 = EC(B) &+ \sum_{k=1}^{N_2} \left( (b_k - a_k) \times \min \left( G_k^-(a_1, \dots, a_{k-1}, b_k, \dots, b_{N_2}), 0 \right) \right)
\end{aligned}$$

C.Q.F.D.

La borne inférieure ( $BI$ ) utilisée dans la PSE est le maximum des deux bornes  $BI_1$  et  $BI_2$ .

En obtenant ces bornes et propriétés de dominance, nous pouvons utiliser le même algorithme PSE que celui proposé dans Ould Louly et Dolgui (2008). Nous rappelons son déroulement dans les deux sections suivantes.

#### 4.3.3.6 Borne supérieure

Comme borne supérieure nous utilisons la première solution trouvée par une recherche en profondeur d'abord. Cette solution est calculée en temps polynomial.

Cette procédure commence par la division de l'espace de recherche en deux fils  $[A, B^1]$  et  $[A^1, B]$ . Cette division du pavé  $[A, B]$  en deux pavés  $[A, B^1]$  et  $[A^1, B]$  est faite en coupant le côté le plus long du pavé  $[A, B]$ .

#### Procédure Borne supérieure

##### Début

//Initialisation de l'espace  $[A, B]$ //

Appliquer limite inférieure sur l'ensemble  $[A, B]$

Appliquer limite supérieure sur l'ensemble  $[A, B]$

//Initialisation de l'espace  $[A, B]$ //

$A \leftarrow (a_1, \dots, a_k, \dots, a_{N_2})$

$B \leftarrow (b_1, \dots, b_k, \dots, b_{N_2})$

//Initialisation de la borne supérieure//

$BS \leftarrow \min(EC(A), EC(B))$

**Tant que** (cardinal de  $[A, B]$  est supérieure à 1) **faire**

Séparer  $[A, B]$  en deux sous ensembles  $[A, B^1]$  et  $[A^1, B]$

Appliquer limite supérieure sur l'ensemble  $[A, B^1]$

Appliquer limite inférieure sur l'ensemble  $[A^1, B]$

$BS \leftarrow$  la solution avec le coût le plus faible, choisie parmi les quatre possibilités :  $A$ ,  $B^1$ ,  $A^1$  ou  $B$ .

**Si**  $(A, B^1)$  **alors**  $[A, B] \leftarrow [A, B^1]$

**Sinon**  $[A, B] \leftarrow [A^1, B]$

**Fin Tant que**

**Fin**

#### 4.3.3.7 Algorithme PSE

L'algorithme PSE que nous proposons utilise pour branchement le schéma suivant. Le sous-ensemble de pavé  $[A, B]$  dont le cardinal est le plus grand est divisé en deux sous-ensembles fils  $[A, B^1]$  et  $[A^1, B]$ . Avant d'être intégré dans l'arbre, chaque sous-ensemble fils est réduit en utilisant les propriétés de dominances. L'idée de base est donc la recherche en largeur d'abord mais avec des modifications en tenant compte des résultats de la réduction.

#### Procédure PSE

**Début**

$BS \leftarrow$  la borne supérieure de  $([A, B])$

$le\_min\_actuel \leftarrow BS$

$La\_solution\_actuelle \leftarrow$  celle donnée par  $BS$

// Initialiser l'ensemble  $E$  des sous ensembles

$E \leftarrow \{ [A, B] \}$

**Si**  $A \neq B$  **alors**

**Tant que** ( $E$  n'est pas vide) **faire**

$[A, B] \leftarrow$  l'élément de  $E$  de cardinal maximal

$E \leftarrow E - \{ [A, B] \}$

Diviser  $[A, B]$  en deux sous-ensembles :  $[A, B^1]$  and  $[A^1, B]$

**Si** (l'ensemble  $[A, B^1]$  n'est pas dominé,  $A \neq B^1$ , sa borne inférieure  $BI < le\_min\_actuel$ ) **alors**

On l'ajoute à  $E$  :  $E \leftarrow E \cup \{ [A, B^1] \}$

**Fin Si**

**Si** ( $A^1 \neq B$  et sa borne inférieure  $BI < le\_min\_actuel$ ) **alors**

On l'ajoute à  $E$  :  $E \leftarrow E \cup \{ [A^1, B] \}$

**Fin Si**

**Si** ( $A^1$  est une solution réalisable et  $C(A^1) < le\_min\_actuel$ ) **alors**

$le\_min\_actuel \leftarrow C(A^1)$

$La\_solution\_actuelle \leftarrow A^1$

**Fin si**

**Si** ( $B^1$  est une solution réalisable et  $C(B^1) < le\_min\_actuel$ ) **alors**

$le\_min\_actuel \leftarrow C(B^1)$

$La\_solution\_actuelle \leftarrow B^1$

**Fin si**

Eliminer les éléments de  $E$  dont la borne inférieure ( $BI$ ) est plus grande que  $le\_min\_actuel$ .

**Fin Tant que**

**Fin si**

**Fin**

Nous présentons dans ce qui suit un exemple numérique qui montre l'utilité de ce pré-traitement de réduction de l'espace de recherche de solutions.

#### 4.3.4 Exemples Numériques

##### 4.3.4.1 Impact de la pré-traitement pour la réduction de l'espace de recherche

Afin d'illustrer la procédure de réduction de l'espace de recherche des solutions, nous présentons un exemple de système d'assemblage à deux niveaux avec deux types de composants au niveau 1 et 10 types des composants au niveau 2 (5 d'entre eux sont destinés à l'assemblage du premier composant de niveau 1 et les 5 autres du deuxième, i.e.  $|S_{1,1}|=5$  et  $|S_{2,1}|=5$ ). La probabilité de distribution des délais d'approvisionnement est donnée dans le tableau 4.1

**Table 4.1 :** La probabilité de distribution des délais d'approvisionnement

$w$	1	2	3	4	5
Pr ( $L_{1,1}=w$ )	0,50	0,30	0,10	0,05	0,05
Pr ( $L_{2,1}=w$ )	0,20	0,20	0,20	0,20	0,20
Pr ( $L_{1,2}=w$ )	0,15	0,30	0,20	0,15	0,20
Pr ( $L_{2,2}=w$ )	0,10	0,20	0,30	0,15	0,25
Pr ( $L_{3,2}=w$ )	0,40	0,10	0,20	0,15	0,15
Pr ( $L_{4,2}=w$ )	0,40	0,30	0,10	0,10	0,10
Pr ( $L_{5,2}=w$ )	0,20	0,20	0,20	0,20	0,20
Pr ( $L_{6,2}=w$ )	0,20	0,20	0,25	0,15	0,20
Pr ( $L_{7,2}=w$ )	0,15	0,20	0,50	0,05	0,10
Pr ( $L_{8,2}=w$ )	0,10	0,10	0,20	0,30	0,30
Pr ( $L_{9,2}=w$ )	0,20	0,25	0,10	0,05	0,05
Pr ( $L_{10,2}=w$ )	0,10	0,35	0,15	0,35	0,05

Le coût de rupture unitaire de produit fini est égal à :  $b = 100$ . Les coûts de stockages unitaires au niveau 1 et au niveau 2 sont présentés dans le tableau 4.2.

**Table 4.2 :** Les coûts de stockages unitaires

$w$	1	2	3	4	5	6	7	8	9	10
Niveau 1 : $h_{w,1}$	60	50								
Niveau 2 : $h_{w,2}$	40	30	20	10	15	25	18	30	25	10

En première étape, nous résolvons le problème  $E_0$  en utilisant la PSE la solution obtenue est :

$$(x_{1,1}^*, x_{2,1}^*) = (2, 4).$$

Le coût de rupture maximal pour les problèmes  $E_i$ ,  $i=1, 2$  est :  $f(y) = 160$ .

Donc, nous résolvons ces 2 problèmes  $E_i$  en appliquant deux valeurs de coût de rupture : i)  $b=0$ , et ii)  $b=160$ .

Les limites inférieures et supérieures des variables de décision  $X_{k,2}$ , ç.-à.-d. les vecteurs  $A$  et  $B$  calculés en utilisant l'approche proposée, pour la réduction de l'espace de recherche, sont présentés dans le tableau 4.3.

**Tableau 4.3 :** Les limites inférieures et supérieures

$k$	1	2	3	4	5	6	7	8	9	10
$a_k$	4	5	5	5	5	6	5	6	6	6
$b_k$	7	7	7	7	7	9	9	9	9	9

Ces limites inférieures et supérieures réduisent énormément l'espace de recherche des solutions qui était initialement :  $a_k = 2$  et  $b_k = 10$  et qui est devenue maintenant  $A=(4,5,5,5,5,6,5,6,6,6)$  et  $B=(7,7,7,7,7,9,9,9,9,9)$ .

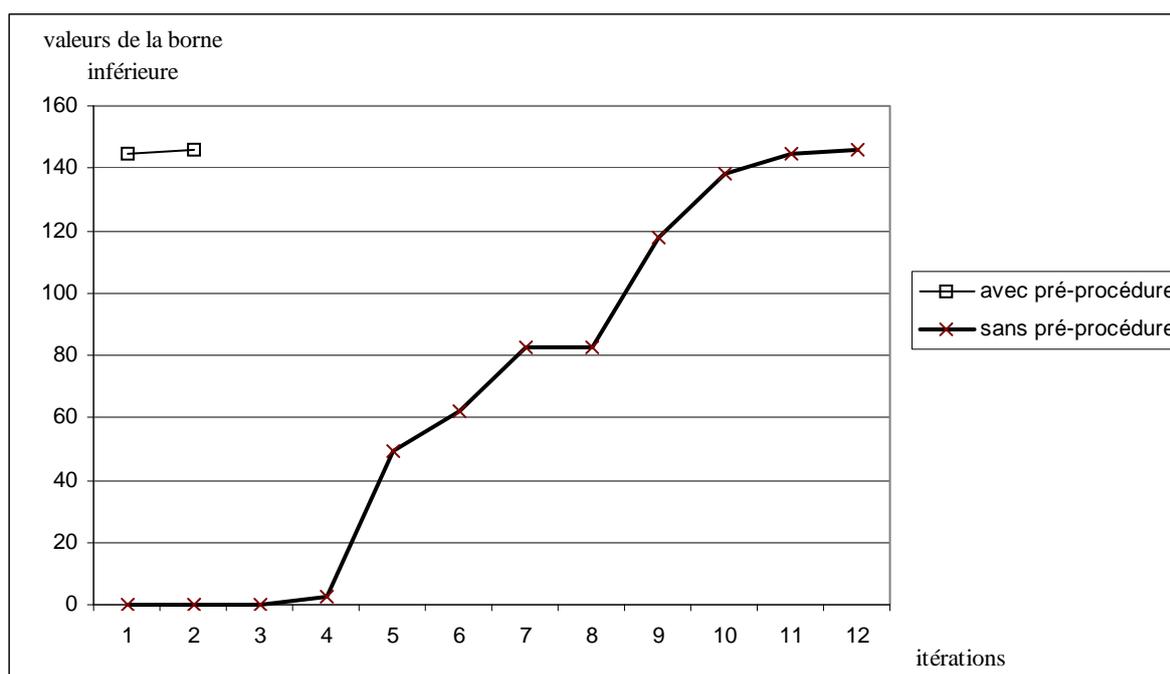
#### 4.3.4.2 Comportement de PSE

Nous avons utilisé un exemple avec  $N_1=2$ ,  $N_2=10$  types de composants (au deuxième niveau 5 composants sont nécessaires pour l'assemblage de chaque composant de niveau 1). Les délais de livraison dont la réalisation maximale  $u_{i,j}$  est égale à 5. Le cardinal de l'espace de recherche initiale est donc égal à  $9^{10}$  ( $U^{N_2}$ ) solutions. Les coûts de stockage unitaire et le coût de rupture unitaire ont été générés d'une manière aléatoire. Les délais d'approvisionnement sont aussi générés d'une manière aléatoire. Cette probabilité de distribution est donnée dans le tableau 4.4.

**Table 4.4 :** La probabilité de distribution des délais d'approvisionnement

$w$	1	2	3	4	5
Pr ( $L_{1,1}=w$ )	0,310453	0,076504	0,559351	0,046432	0,007260
Pr ( $L_{2,1}=w$ )	0,103505	0,102381	0,220563	0,259308	0,314243
Pr ( $L_{1,2}=w$ )	0,000582	0,261902	0,089830	0,375828	0,271858
Pr ( $L_{2,2}=w$ )	0,117162	0,299672	0,275215	0,249717	0,058234
Pr ( $L_{3,2}=w$ )	0,434712	0,314200	0,185996	0,009168	0,055924
Pr ( $L_{4,2}=w$ )	0,078924	0,088882	0,529611	0,238783	0,063800
Pr ( $L_{5,2}=w$ )	0,004261	0,180683	0,254214	0,273111	0,287731
Pr ( $L_{6,2}=w$ )	0,098408	0,392513	0,266861	0,208452	0,033766
Pr ( $L_{7,2}=w$ )	0,238545	0,244419	0,158321	0,091953	0,266762
Pr ( $L_{8,2}=w$ )	0,315959	0,305983	0,178276	0,047048	0,152734
Pr ( $L_{9,2}=w$ )	0,233546	0,056773	0,211178	0,228512	0,269991
Pr ( $L_{10,2}=w$ )	0,253990	0,163001	0,110573	0,123477	0,348959

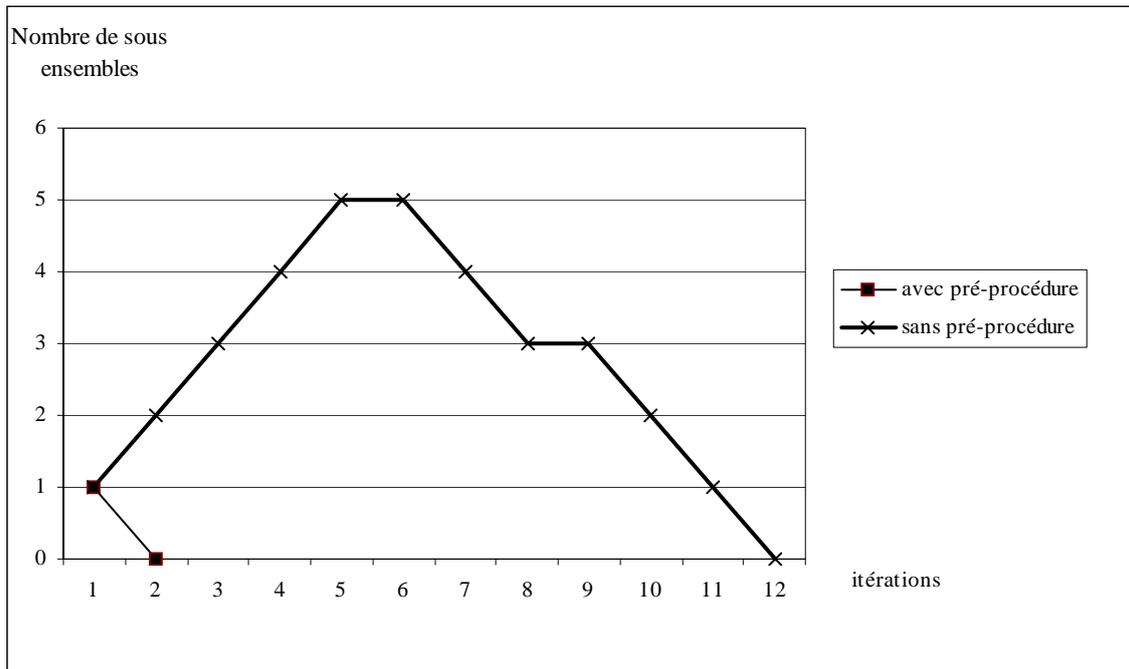
Nous avons obtenu la valeur optimale de la fonction objectif est égal à 145,8170. Nous l'avons obtenu après 12 itérations sans pré-traitement de réduction de l'espace de recherche de solution et après uniquement 2 itérations avec le pré-traitement de réduction. L'évolution de la borne inférieure obtenue au cours des itérations avec et sans pré-traitement est présentée dans la figure 4.4.



**Figure 4.4 :** L'évolution de la borne inférieure pour le problème PP1

L'évolution du nombre de sous-ensembles est présentée dans la figure 4.5. Au début lorsque la borne inférieure n'est pas encore trop précise, le nombre de sous-ensembles augmente jusqu'à 5 pour le cas sans pré-traitement de réduction de l'espace de recherche des solutions. Elle commence à 1 pour le cas

avec ce pré-traitement. Dans la deuxième phase, il décroît pour enfin s'annuler après 12 itérations (2 itérations sont nécessaires pour l'annuler dans le cas avec le pré-traitement).



**Figure 4.5 :** L'évolution de nombre des sous-ensembles d'arbre de recherche pour le problème PP1

#### 4.3.4.3 Tests des performances de l'algorithme

Les testes consistaient à exécuter l'algorithme sur 100 instances générées d'une manière aléatoire. Pour ces instances, le nombre des composants au niveau 2 était choisi parmi les valeurs suivantes [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]. Pour chaque nombre des composants, 10 instances différentes sont générées. Les données de chaque instance sont : la fonction de distribution des délais d'approvisionnements des composants (le nombre de réalisation maximale=5), les coûts unitaires de stockage des composants, le coût unitaire de rupture en produit fini et le nombre des composants au niveau 1 nécessaires pour l'assemblage de produit fini et le nombre des composants de niveau 2 nécessaires pour l'assemblage de chaque type des composants de niveau 1.

Nous présentons les résultats numériques pour 100 tests générés d'une manière aléatoire.

En utilisant la PSE proposé précédemment sans le pré-traitement pour la réduction de l'espace de recherche des solutions, les instances de taille supérieure à 30 n'ont pas été résolues dans un temps qui a été alloué d'une heure. Lorsque, nous avons ajouté le pré-traitement pour la réduction de recherche des solutions, nous nous sommes arrivés à résoudre les instances jusqu'à taille 80. Cependant les instances avec 90 et 100 composants n'ont pas été résolues dans le temps alloué vu le cardinal très important de l'espace de recherche dans ce cas - là (voir le tableau 4.5).

**Table 4.5** : Le temps moyen de calcul (en seconds)

$N_2$	10	20	30	40	50	60	70	80	90	100
PSE Sans pré-traitement	0,063	1,234	11,261	Pas résolu						
PSE Avec pré-traitement	0	0,160	0,719	5,700	9,500	1,859	11,100	14,950	Pas résolu	

Les résultats montrent l'efficacité des algorithmes proposés pour les problèmes de petite taille et de taille moyenne. La résolution optimale du problème dépend donc fortement de la qualité du pré-traitement pour la réduction de l'espace de recherche initiale et de la qualité de la PSE en terme de propriétés de dominance, borne inférieure et borne supérieure.

## 4.4 Problème PP2

Nous allons maintenant traiter un problème similaire mais avec un niveau de service objectif. Le problème est donc de trouver la planification optimale de ce type de système d'assemblage en minimisant le coût engendré par le stockage des produits de niveau 1 et de niveau 2 tout en respectant le niveau de service désiré par le client final. Dans ce qui suit, nous présentons analytiquement le coût total moyen à minimiser.

### 4.4.1 Description du problème

Comme ici, nous considérons le cas d'assemblage au plus tôt : un produit est assemblé dès que tous les composants nécessaires pour son assemblage sont présents, le problème est donc de trouver les dates de lancement optimales au niveau 2 afin de satisfaire la demande  $D$  à la date souhaitée  $T$ , en minimisant le coût total moyen (voir la figure 4.6).

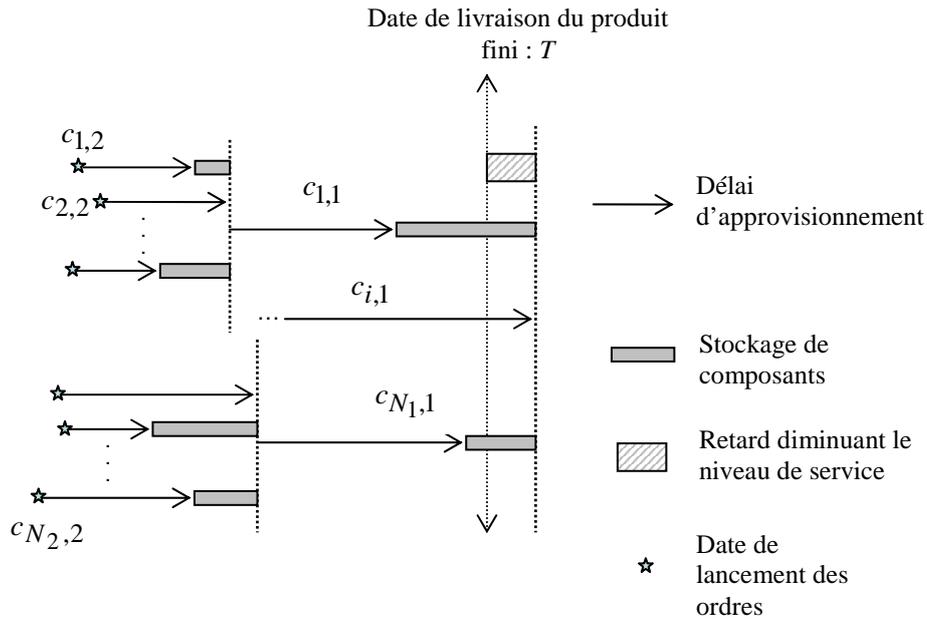


Figure 4.6 : Description du problème PP2

#### 4.4.2 Minimisation des coûts

**Proposition 4.10** Pour ce système d'assemblage à deux niveaux, le coût total du PP2 s'exprime comme suit :

$$C(X, L) = H \max_{i=1, \dots, N_1} (M_{i,1} + L_{i,1})^+ - \sum_{i=1}^{N_1} h_{i,1} (M_{i,1} + L_{i,1}) + \sum_{i=1}^{N_1} \left( \sum_{c_{k,2} \in S_{i,1}} h_{k,2} (M_{i,1} - (L_{k,2} - X_{k,2})) \right) \quad (4.33)$$

sous la contrainte :

$$\Pr \left( \max_{i=1, \dots, N_1} (M_{i,1} + L_{i,1})^+ > 0 \right) \leq \varepsilon \quad (4.34)$$

où,

$$L = (L_{1,1}, \dots, L_{N_1,1}; L_{1,2}, \dots, L_{N_2,2})$$

$$X = (X_{1,2}, \dots, X_{k,2}, \dots, X_{N_2,2})$$

$$M_{i,1} = \max_{c_{k,2} \in S_{i,1}} (L_{k,2} - X_{k,2})$$

$$H = \sum_{i=1}^{N_1} h_{i,1}$$

### Démonstration

L'objectif est de réduire le coût de stockage total (4.33). Cette fonction s'obtient à partir de (4.1) en mettant  $b=0$ .

Le coût de rupture  $b \max_{i=1, \dots, N_1} (M_{i,1} + L_{i,1})^+$  est remplacé par la contrainte sur le niveau de service, la probabilité de rupture ne doit donc pas dépasser la probabilité de rupture tolérée :

$$\Pr\left(\max_{i=1, \dots, N_1} (M_{i,1} + L_{i,1})^+ > 0\right) \leq \varepsilon.$$

C.Q.F.D.

Les délais d'approvisionnement  $L_{i,j}$  sont des variables aléatoires et donc le coût ci-dessus (4.33) est une variables aléatoire. Notre objectif est de minimiser l'espérance mathématique (4.33), c'est-à-dire le coût total moyen  $EC(X)$ .

**Proposition 4.11** *Le coût total moyen du PP2 s'exprime comme suit :*

$$\begin{aligned} EC(X) = & H \sum_{s \geq 0} \left( 1 - \prod_{i=1}^{N_1} \left( \sum_{o_1+o_2=s} \Pr(L_{i,1} = o_1) \times \prod_{c_{k,2} \in S_{i,1}} F_{k,2}(X_{k,2} + o_2) \right) \right) \\ & + \sum_{i=1}^{N_1} H_i \times \left( \sum_{s \geq 0} \left( 1 - \prod_{c_{k,2} \in S_{i,1}} F_{k,2}(X_{k,2} + s) \right) \right) \\ & - \sum_{i=1}^{N_1} H_i \times \left( \sum_{s \geq 0} \left( 1 - \prod_{c_{k,2} \in S_{i,1}} 1 - F_{k,2}(X_{k,2} - s - 1) \right) \right) \\ & - \sum_{i=1}^{N_1} \left( \sum_{c_{k,2} \in S_{i,1}} h_{k,2} (E(L_{k,2}) - X_{k,2}) \right) - \sum_{i=1}^{N_1} h_{i,1} E(L_{i,1}) \end{aligned} \quad (4.35)$$

$$\text{sous la contrainte : } \prod_{i=1}^{N_1} \left( \sum_{s \geq 0} \Pr(L_{i,1} = s) \times \prod_{c_{k,2} \in S_{i,1}} (F_{k,2}(X_{k,2} - s)) \right) \geq 1 - \varepsilon \quad (4.36)$$

où,

$$H = \sum_{i=1}^{N_1} h_{i,1}, \quad H_i = \sum_{c_{k,2} \in S_{i,1}} (h_{k,2}) - h_{i,1}, \quad \text{pour } i = 1, \dots, N_1$$

$$X = (X_{1,2}, \dots, X_{i,2}, \dots, X_{N_2})$$

### Démonstration

Le coût de stockage total moyen (4.35) a été démontré dans (4.5), il suffit de remplacer  $b$  par 0. Le coût de rupture  $b \max_{i=1, \dots, N_1} (M_{i,1} + L_{i,1})^+$  est donc remplacé par la contrainte sur le niveau de service :

$$\begin{aligned} \Pr\left(\max_{i=1, \dots, N_1} (M_{i,1} + L_{i,1})^+ > 0\right) &= 1 - \Pr\left(\max_{i=1, \dots, N_1} (M_{i,1} + L_{i,1})^+ \leq 0\right) = 1 - \prod_{i=1}^{N_1} (M_{i,1} + L_{i,1} \leq 0) \\ &= 1 - \prod_{i=1}^{N_1} \left(\max_{c_{k,2} \in \mathcal{S}_{i,1}} (L_{k,2} - X_{k,2}) + L_{i,1} \leq 0\right) \\ &= 1 - \prod_{i=1}^{N_1} \left(\sum_{s \geq 0} \Pr(L_{i,1} = s) \times \prod_{c_{k,2} \in \mathcal{S}_{i,1}} (F_{k,2}(X_{k,2} - s))\right) \end{aligned}$$

C.Q.F.D.

### 4.4.3 Optimisation

#### 4.4.3.1 Problèmes à résoudre

Notre problème peut donc se modéliser comme suit :

#### Problème PP2 :

Min  $EC(X)$

Sous les contraintes :

(4.6), (4.7), (4.8), (4.9) et

$$\prod_{i=1}^{N_1} \left(\sum_{s \geq 0} \Pr(L_{i,1} = s) \times \prod_{c_{k,2} \in \mathcal{S}_{i,1}} (F_{k,2}(X_{k,2} - s))\right) \geq 1 - \varepsilon \quad (4.37)$$

#### 4.4.3.2 Pré-traitement pour la réduction de l'espace de recherche

D'après Ould Louly *et al.* (2008 b) pour le cas du problème de minimisation de coût moyen sous contrainte de niveau de service, le problème peut être optimisé par une PSE similaire à celle du problème où le critère de niveau de service est remplacé par un coût de rupture.

Soit  $EC_0(x_{1,1}^*, \dots, x_{i,1}^*, \dots, x_{N_1,1}^*)$  le coût moyen de la solution optimal de  $E_0$ .

Avec :

$$EC_0(x_{1,1}^*, \dots, x_{i,1}^*, \dots, x_{N_1,1}^*) = \sum_{i=1}^{N_1} h_{i,1} (x_{i,1}^* - E(L_{i,1})) + H \sum_{k \geq 0} \left( 1 - \prod_{i=1}^{N_1} F_{i,1}(x_{i,1}^* + k) \right) \quad (4.38)$$

Sous la contrainte de niveau de service :

$$\prod_{i=1}^{N_1} F_{i,1}(x_{i,1}^* + k) \geq 1 - \varepsilon \quad (4.39)$$

où,

$$H = \sum_{i=1}^{N_1} h_{i,1}$$

$\varepsilon \in [0, 1]$  est la probabilité de rupture tolérée (le niveau de service exigé est alors égal à  $1 - \varepsilon$ ).

Comme les limites inférieures et les limites supérieures des variables  $x_{k,2}$ ,  $k=1, \dots, N_2$ , sont respectivement 1 et  $u_{k,2}$ , nous pouvons calculer les limites inférieures et supérieures des variables de décision  $X_{k,2}$  comme suit :

$$a_k = x_{i,1}^* + 1 \quad (4.40)$$

$$b_k = x_{i,1}^* + u_{k,2} \quad (4.41)$$

pour  $k=1, 2, \dots, N_2$ ;  $i=1, 2, \dots, N_1$ , et  $c_{k,2} \in S_{i,1}$ .

En plus, chaque solution  $X_{k,2}$  doit vérifier la contrainte suivante :

$$LL_{k,2} \leq X_{k,2} \leq U_{k,2}, \quad k=1, \dots, N_2. \quad (4.42)$$

où  $LL_{k,2}$  est le plus petit entier qui vérifie :

$$F_{k,2}(X_{k,2}) \geq 1 - \varepsilon \quad (4.43)$$

En effet, un vecteur  $X$  qui a une composante  $X_{k,2}$ , plus petit que  $LL_{k,2}$ , n'est pas une solution réalisable, car la contrainte sur le niveau de service du problème PP2 (la relation (4.9) n'est pas respectée :

$$\prod_{i=1}^{N_1} \left( \sum_{s \geq 0} \Pr(L_{i,1} = s) \prod_{c_{k,2} \in S_{i,1}} (F_{k,2}(X_{k,2} - s)) \right) \leq \prod_{i=1}^{N_1} \prod_{c_{k,2} \in S_{i,1}} (F_{k,2}(X_{k,2})) \leq F_{k,2}(X_{k,2}) \leq 1 - \varepsilon$$

Donc les limites inférieures et supérieures des variables de décision  $X_{k,2}$  peuvent être réécrites comme suit :

$$a_k = \max(x_{i,1}^* + 1, LL_{k,2}) \quad (4.44)$$

$$b_k = x_{i,1}^* + u_{k,2} \quad (4.45)$$

pour  $k= 1,2,\dots, N_2$ ;  $i= 1,2,\dots, N_1$ , et  $c_{k,2} \in S_{i,1}$ .

Une particularité du problème PP2 est que l'ensemble initial  $[A, B]$  avec  $A = (a_1, \dots, a_k, \dots, a_{N_2})$  et  $B = (b_1, \dots, b_k, \dots, b_{N_2})$  peut contenir des solutions qui ne sont pas réalisables. En effet, le vecteur de cet ensemble qui donne le niveau de service le plus faible est le vecteur  $A = (a_1, \dots, a_k, \dots, a_{N_2})$ . D'après la définition de  $LL_{k,2}$ , ce vecteur vérifie l'expression suivante :

$$\begin{aligned} \prod_{i=1}^{N_1} \left( \sum_{s \geq 0} \Pr(L_{i,1} = s) \prod_{c_{k,2} \in S_{i,1}} (F_{k,2}(X_{k,2} - s)) \right) &\geq \prod_{i=1}^{N_1} \left( \sum_{s \geq 0} \Pr(L_{i,1} = s) \times (1 - \varepsilon)^{n_{i,1}} \right) \\ &\geq \prod_{i=1}^{N_1} (1 - \varepsilon)^{n_{i,1}} \geq (1 - \varepsilon)^{N_2} \end{aligned} \quad (4.46)$$

mais il ne garantit pas dans le cas général le respect de la contrainte initiale sur le niveau de service :

$$\prod_{i=1}^{N_1} \left( \sum_{s \geq 0} \Pr(L_{i,1} = s) \prod_{c_{k,2} \in S_{i,1}} (F_{L_{k,2}}(X_{k,2} - s)) \right) \geq (1 - \varepsilon) \quad (4.47)$$

Remarquons que le niveau de service est une fonction croissante par rapport à chaque variable  $X_{k,2}$ .

Un ensemble  $[A, B]$  tel que le vecteur  $B$  ne respecte pas la contrainte sur le niveau de service ne contiendra aucune solution réalisable, et peut donc être éliminé de l'espace de recherche.

#### 4.4.3.3 Propriétés de dominance

**Proposition 4.11** *Le problème PP2 a les deux propriétés suivantes :*

(i) *si  $\exists k$  tel que  $G_k^+(A) < 0$ , alors chaque solution  $X$  de  $[A, B]$  avec  $X_{k,2} = a_k$  est dominée.*

(ii) *si  $\exists k$  tel que  $G_k^-(B) < 0$  et tel que le vecteur  $(a_1, \dots, a_{k-1}, b_k - 1, \dots, a_{N_2})$  vérifie la contrainte de niveau de service, alors chaque solution  $X$  de  $[A, B]$  avec  $X_{k,2} = b_k$  est dominée.*

### Démonstration

La démonstration de (i) est exactement la même que celle de la proposition 4.10. Comme le niveau de service est une fonction croissante par rapport à chaque variable  $X_{k,2}$ , alors le vecteur  $(X_{1,2}, \dots, X_{k-1,2}, a_k + 1, X_{k+1,2}, \dots, X_{N_2,2})$  qui donne un coût plus faible que le coût de  $(X_{1,2}, \dots, X_{k-1,2}, a_k, X_{k+1,2}, \dots, X_{N_2,2})$ , donne un plus grand niveau de service.

La propriété (ii) se démontre de la même manière que pour la proposition 4.8. En effet, le vecteur  $(X_{1,2}, \dots, X_{k-1,2}, b_k, X_{k+1,2}, \dots, X_{N_2,2})$  sera dominé par le vecteur  $(X_{1,2}, \dots, X_{k-1,2}, b_k - 1, X_{k+1,2}, \dots, X_{N_2,2})$ . Ce dernier est une solution réalisable, car le vecteur  $(a_1, \dots, a_{k-1}, b_k - 1, \dots, a_{N_2})$ , dont le niveau de service est plus faible, déjà une solution réalisable.

C.Q.F.D.

Il résulte de ces propriétés, de la même manière que pour le problème PP1, que si  $\exists k$  tel que  $G_k^+(A) < 0$  et  $a_k < b_k$ , alors l'ensemble des solutions  $[A, B]$  peut être remplacé par son sous ensemble où  $a_k$  est remplacé par  $a_k + 1$ . Et si  $\exists k$  tel que  $G_k^+(B) < 0$  et que le vecteur  $(a_1, \dots, a_{k-1}, b_k - 1, \dots, a_{N_2})$  vérifie la contrainte de niveau de service et que  $a_k < b_k$ , alors l'ensemble des solutions  $[A, B]$  peut être remplacé par son sous ensemble où  $b_k$  est remplacé par  $b_k - 1$ .

Les propriétés de dominance peuvent être utilisées pour développer des procédures efficaces de coupe pour la PSE. En effet, après la division d'un noeud, dans une PSE, deux noeuds (descendants) sont créés. Pour chaque noeud, les propriétés de dominance sont utilisées pour réduire les espaces de recherche correspondants avant de passer au prochain branchement.

Les descendants de l'espace  $[A, B]$  sont obtenues en partageant cet ensemble en deux sous-ensembles  $[A, B^1]$  et  $[A^1, B]$  pour respectivement  $a_k \leq X_{k,2} \leq \frac{a_k + b_k}{2}$  et  $\frac{a_k + b_k}{2} + 1 \leq X_{k,2} \leq b_k$ .

La procédure de coupe inférieure du problème PP1 reste valable pour le problème PP2. Ce n'est pas le cas pour la procédure de coupe supérieure du problème PP1, à sa place nous proposons d'utiliser la procédure de limite supérieure suivante (voir Ould Louly *et al.*, 2008 b)

### **Procédure de coupe supérieure PP2**

#### **Début**

réduit  $\leftarrow$  faux

dominé  $\leftarrow$  faux

**Tant que** (non réduit et non dominé)

réduit  $\leftarrow$  vrai

**Pour**  $k = 1$  à  $N_2$  **faire**

**Tant que**

$$\left( G_k^-(B) < 0, \prod_{i=1}^{N_1} \left( \sum_{s \geq 0} \Pr(L_{i,1} = s) F_{k,2}(b_k - 1 - s) \prod_{\substack{c_{j,2} \in \mathcal{S}_{i,1} \\ j \neq k}} (F_{j,2}(a_{j,2} - s)) \right) \geq 1 - \varepsilon \right)$$

et dominé = faux) **faire**

réduit  $\leftarrow$  faux

si  $(a_k = b_k)$  dominé  $\leftarrow$  vrai

sinon  $b_k \leftarrow b_k - 1$

**Fin Tant que**

**Fin Pour**

**Fin Tant que**

**Fin**

#### 4.4.3.4 Bornes inférieures

Nous allons maintenant présenter deux bornes inférieures des fonctions objectifs que nous utiliserons dans nos algorithmes d'optimisation. Ces bornes s'appuient sur l'étude des accroissements partiels des fonctions objectifs (Ould Louly *et al.*, 2008 b).

**Proposition 4.12** Les valeurs  $BI_1$  et  $BI_2$  sont deux bornes inférieures pour la fonction coût sur l'ensemble  $[A, B]$  :

$$BI_1 = EC(A) + \sum_{k=1}^{N_2} \left( (b_k - a_k) \times \min \left( G_k^+(b_1, \dots, b_{k-1}, a_k, \dots, a_{N_2}), 0 \right) \right) \quad (4.41)$$

$$BI_2 = EC(B) + \sum_{k=1}^{N_2} \left( (b_k - a_k) \times \min \left( G_k^-(a_1, \dots, a_{k-1}, b_k, \dots, b_{N_2}), 0 \right) \right) \quad (4.42)$$

#### Démonstration

En relaxant la contrainte de niveau de service pour le problème PP1, la démonstration devient identique pour les deux problèmes PP1 et PP2.

C.Q.F.D.

La borne inférieure ( $BI$ ) utilisée dans la PSE est le maximum des deux bornes  $BI_1$  et  $BI_2$ .

#### 4.4.3.5 Borne supérieure

Comme borne supérieure nous utilisons le même principe du problème PP1.

##### Procédure Borne supérieure PP2

###### Début

//Initialisation de l'espace  $[A, B]$ //

Appliquer limite inférieure sur l'ensemble  $[A, B]$

Appliquer limite supérieure sur l'ensemble  $[A, B]$

//Initialisation de l'espace  $[A, B]$ //

$A \leftarrow (a_1, \dots, a_k, \dots, a_{N_2})$

$B \leftarrow (b_1, \dots, b_k, \dots, b_{N_2})$

//Initialisation de la borne supérieure//

$BS \leftarrow \min(EC(A), EC(B))$

**Tant que** (cardinal de  $[A, B]$  est supérieure à 1) **faire**

    Séparer  $[A, B]$  en deux sous ensembles  $[A, B^1]$  et  $[A^1, B]$

    Appliquer limite supérieure sur l'ensemble  $[A, B^1]$

    Appliquer limite inférieure sur l'ensemble  $[A^1, B]$

$BS \leftarrow$  la solution réalisable avec le coût le plus faible, choisie parmi les quatre possibilités :  $A, B^1, A^1$  ou  $B$ .

**Si**  $(A, B^1)$  **alors**  $[A, B] \leftarrow [A, B^1]$

**Sinon**  $[A, B] \leftarrow [A^1, B]$

**Fin Tant que**

**Fin**

#### 4.4.3.6 Algorithme PSE

L'algorithme PSE que nous proposons utilise le même principe que PP1.

##### Procédure PSE

###### Début

$BS \leftarrow$  la borne supérieure de  $([A, B])$

$le\_min\_actuel \leftarrow BS$

$La\_solution\_actuelle \leftarrow$  celle donnée par  $BS$

// Initialiser l'ensemble  $E$  des sous ensembles

$E \leftarrow \{ [A, B] \}$

**Si**  $A \neq B$  **alors**

**Tant que** ( $E$  n'est pas vide) **faire**

$[A, B] \leftarrow$  l'élément de  $E$  de cardinal maximal

$E \leftarrow E - \{ [A, B] \}$

Diviser  $[A, B]$  en deux sous-ensembles:  $[A, B^1]$  and  $[A^1, B]$

**Si** (l'ensemble  $[A, B^1]$  n'est pas dominé,  $A \neq B^1$ , sa borne inférieure  $BI < le\_min\_actuel$ ) **alors**

On l'ajoute à  $E$  :  $E \leftarrow E \cup \{ [A, B^1] \}$

**Fin Si**

**Si** ( $A^1 \neq B$  et sa borne inférieure  $BI < le\_min\_actuel$ ) **alors**

On l'ajoute à  $E$  :  $E \leftarrow E \cup \{ [A^1, B] \}$

**Fin Si**

**Si** ( $C(A^1) < le\_min\_actuel$ ) **alors**  $le\_min\_actuel \leftarrow EC(A^1)$

$La\_solution\_actuelle \leftarrow A^1$

**Fin si**

**Si** ( $C(B^1) < le\_min\_actuel$ ) **alors**  $le\_min\_actuel \leftarrow EC(B^1)$

$La\_solution\_actuelle \leftarrow B^1$

**Fin si**

Eliminer les éléments de  $E$  dont la borne inférieure (BI) est plus grande que  $le\_min\_actuel$ .

**Fin Tant que**

**Fin si**

**Fin**

#### 4.4.4 Exemples Numériques

##### 4.4.4.1 Impact de la procédure de pré-traitement pour la réduction de l'espace de recherche

En première étape, nous résolvons le problème  $E_0$  en utilisant la PSE proposée. La solution obtenue est :

$$(x_{1,1}^*, x_{2,1}^*) = (4, 5).$$

En utilisant (4.44) et (4.45), nous obtenons les limites inférieures et supérieures des variables de décision.

**Table 4.6 :** Les limites inférieures et supérieures des variables de décision

$k$	1	2	3	4	5	6	7	8	9	10
$a_k$	5	5	5	5	5	6	6	6	6	6
$b_k$	9	9	9	9	9	10	10	10	10	10

Ces limites inférieures et supérieures, comme le problème PP1, réduisent énormément l'espace de recherche des solutions qui était initialement :  $a_k = 2$  et  $b_k = 10$ .

Afin d'évaluer l'effet de ce pré-traitement, nous avons généré 100 scénarios d'une manière aléatoire (un scénario correspond à une loi de distribution de  $L_j, j=1, \dots, 10$ , et à des coûts de stockage  $h_{i,j}$ ) sur la même structure de chaîne logistique avec 2 types de composants au niveau 1 et avec 10 types composants au niveau 2. Les résultats ont montré que l'espace de recherche des solutions a été réduit de  $9^{10}$  ( $U^{N^2}$ )=3486784401 à 976299632 en moyenne. L'espace de recherche des solutions est donc réduit de ~3,5 fois.

#### **4.4.4.2 Etude numérique de l'algorithme**

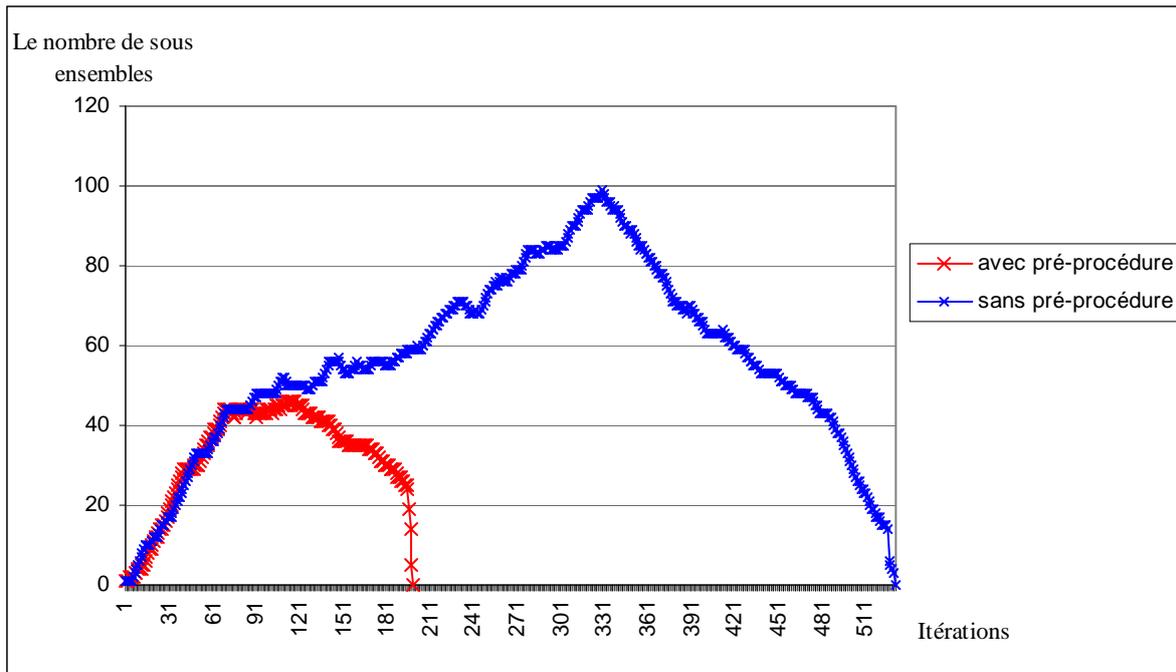
Nous allons maintenant présenter les résultats de testes de la PSE. Un ensemble de tests a été proposé. Nous présentons pour chaque exemple l'évolution de la borne inférieure  $BI$  de l'espace de recherche. Cette borne est calculée pour chaque sous-ensemble. Nous présentons également l'évolution du nombre des ensembles et du cardinal de l'espace de recherche durant le déroulement de l'algorithme d'optimisation. Ces exemples sont suivis de tableaux statistiques donnant des résultats récapitulatifs sur un grand nombre de tests.

#### **4.4.4.3 Comportement de PSE**

Nous avons utilisé le même exemple du problème PP1 décrit dans la section 4.3.4.2 avec un niveau de service désiré égal à 0.95.

Nous avons obtenu les valeurs optimales après 603 itérations sans pré-traitement de réduction (199 itérations ont été nécessairement avec le pré-traitement de réduction). Le minimum est égal à 189,62 et le niveau de service associé est égal à 0,956600.

Dans la figure 4.7, nous présentons l'évolution du nombre de noeuds, elle a deux phases bien distinctes, le nombre de noeuds augmentent jusqu'à 99 sans pré-traitement de réduction (46 avec le pré-traitement) pendant les 329 premières itérations (avec le pré-traitement 117 itérations) et décroît ensuite.



**Figure 4.7 :** L'évolution de nombre des noeuds de l'arbre de recherche pour le problème PP2

#### 4.4.4.4 Tests des performances des algorithmes

Les testes consistaient à exécuter l'algorithme sur 100 instances générées d'une manière aléatoire. Pour ces instances, le nombre des composants au niveau 2 était choisi parmi les valeurs suivantes [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]. Pour chaque nombre des composants, 10 instances différentes sont générées. Les données de chaque instance sont : la fonction de distribution des délais d'approvisionnements des composants (le nombre de réalisation maximale=5), les coûts unitaires de stockage des composants, le coût unitaire de rupture en produit fini et le nombre des composants au niveau 1 nécessaires pour l'assemblage de produit fini et le nombre des composants de niveau 2 nécessaires pour l'assemblage de chaque type des composants de niveau 1.

Nous présentons les résultats numériques pour ces 100 tests dans le tableau 4.7.

En utilisant la PSE sans pré-traitement de réduction de l'espace de recherche, les instances de taille supérieure à 60 n'ont pas été résolues dans un temps qui a été alloué (une heure). Lorsque, nous avons ajouté le pré-traitement de réduction de recherche des solutions, nous ne sommes pas arrivé non plus à résoudre les instances de taille supérieure à 60 dans le temps alloué vu le cardinal très important d'espace de recherche des solutions.

**Table 4.7** : Le temps moyen de calcul (en seconds)

$N_2$	10	20	30	40	50	60	70	80	90	100
PSE Sans pré-traitement	0.313	1.0470	13.79	108.279	199.059	105.935	Pas résolus			
PSE Avec pré-traitement	0.141	0.969	4.094	116.656	104.591	116.233				

## 4.5 Conclusions

Dans ce chapitre, nous avons étudié l’approvisionnement en composants pour les systèmes d’assemblage à deux niveaux avec un seul type de produit fini. Pour ce problème, notre objectif était de trouver la planification optimale. Nous avons cherché les valeurs des dates de lancement des ordres au niveau 2. Nous avons traité deux modèles. Le premier utilise comme critère la somme du coût moyen de stockage des composants et du coût moyen de rupture en produit fini. Le deuxième minimise le coût moyen de stockage des composants sous la contrainte d’un niveau de service.

Nous avons modélisé les critères avec des expressions explicites en fonctions des dates de lancement des ordres. Nous avons ainsi obtenu deux fonctions objectifs non linéaires à variables entiers. Nous avons développé un pré-traitement pour la réduction de l’espace de recherche qui peut être appliquée avant de commencer l’optimisation. Pour chacun de deux problèmes étudiés nous avons montré comment une PSE peut être utilisée. Pour chacune de ces procédures nous avons démontré des propriétés de dominances, des bornes inférieures et une borne supérieure.

Nous avons aussi étudié la performance des algorithmes proposés, les tests ont montré l’importance de cette optimisation et la capacité de ces algorithmes à traiter des problèmes de petite taille et de taille moyenne dans un temps raisonnable.

Dans le chapitre suivant, nous présentons une méta-heuristique basée sur un algorithme génétique qui permet de résoudre les problèmes de taille plus importante dans un temps très réduits tout en maintenant la qualité de la solution à un niveau acceptable.

# Chapitre 5 : Optimisation d'un système d'assemblage à deux niveaux par un AG

## 5.1 Introduction

Dans le chapitre précédent, nous avons présenté une méthode d'optimisation exacte fondée sur une procédure par Séparation et Évaluation (PSE) dont l'objectif est de minimiser l'espérance mathématique de coût total pour les problèmes considérés. Mais les résultats expérimentaux ont montré que elle ne pouvait s'appliquer qu'à des problèmes de petite taille (au maximum 60 variables de décision). En effet, les techniques d'exploration et de coupe de l'espace de recherche ont malheureusement leurs limites face à cette classe de problèmes, ce qui peut engendrer des temps d'exécution assez importants.

Dans ce chapitre, nous proposons donc une étude complémentaire en développant une méthode d'optimisation approchée dont l'objectif est de pouvoir traiter des problèmes de taille importante. L'approche proposée est une méta-heuristique à base d'algorithme génétique (AG) qui permet d'assurer un bon compromis entre la qualité de la solution et le temps de résolution.

Parmi le grand nombre de méta-heuristiques présentées dans la littérature, les algorithmes génétiques semblent bien adaptés à notre problème, car la représentation des solutions et les opérateurs de reproduction (croisement et mutation) peuvent facilement être définis ici. En outre, dans la mesure où il n'existe pas de contraintes particulières, une méta-heuristique basée sur une recherche locale, telle que la recherche Tabou par exemple, demande l'exploration d'un grand nombre de voisins. Cela pourra consommer trop de temps, notamment parce que le calcul de chaque fonction objectif a besoin d'un temps significatif.

Les algorithmes génétiques sont des algorithmes de recherche de solutions basés sur le mécanisme de la sélection naturelle. Ils ont été initialement proposés par Holland (1975) et Golberg (1989). Comme l'ont mentionné Michalewicz et Fogel (2002), les algorithmes génétiques sont largement utilisés sur une très large gamme de problèmes. En particulier, ils ont été appliqués avec succès sur plusieurs problèmes tels que le problème de planification d'un système d'assemblage dont l'objectif est de

minimiser le temps de cycle, le temps de changement d'outils, le nombre de machine et la complexité de séquence d'assemblage (Chen *et al.*, 2002), le dimensionnement des stocks tampon dans une ligne de production (Dolgui *et al.*, 2002, 2007), la gestion des approvisionnements (Borisovsky *et al.*, 2008) la gestion des stocks tel que Sudhir Ryan Daniel et Rajendran, (2005) et Pasandideh et Niaki (2008), l'optimisation de réseaux de chaînes logistiques (Ding *et al.*, 2006) et le problème d'ordonnancement d'un flow shop ou un open shop (Chen *et al.*, 1995, Prins *et al.*, 2000, Wang *et al.*, 2006). Houpt et Houpt (1998) fournissent un excellent aperçu sur les algorithmes génétiques (discrets et continus) et leurs applications. Habituellement, la raison du choix des algorithmes génétiques est leur excellente performance pour des problèmes similaires, et la possibilité d'obtenir plusieurs solutions en même temps.

Nous commençons ce chapitre en présentant le principe général et les différentes étapes des algorithmes génétiques. Nous présentons ensuite la mise en œuvre de l'algorithme génétique sur notre problème. Ensuite, nous présentons les résultats numériques et les comparaisons avec d'autres heuristiques. Afin d'aboutir à un meilleur résultat tant pour la qualité des solutions que pour le temps d'exécution, nous avons mis en place une procédure de recherche locale qui permet d'accélérer la convergence de l'algorithme. Une comparaison avec d'autres heuristiques est également présentée.

## 5.2 Algorithme génétique

Nous étudions au cours de chapitre le même système d'assemblage à deux niveaux que celui décrit dans le chapitre précédent. Nous nous intéressons au problème PP1 du chapitre précédent où le critère est la minimisation de l'espérance mathématique du coût total composé du coût de rupture en produit fini et du coût de stockage des composants. Les définitions et les notations déjà utilisées restent valables ici. Les nouvelles notations concernant les paramètres de l'algorithme génétique seront introduites au fur et à mesure tout au long de la description de l'algorithme.

Ici, l'algorithme génétique est mis en œuvre dans le but de chercher les meilleures dates de lancement des ordres  $X_{k,2}$ ,  $k=1, 2, \dots, N_2$ , afin de minimiser l'espérance mathématique de coût total  $EC(X)$  présenté dans le chapitre précédent :

$$\begin{aligned} \text{Min } EC(X) = & H \times \sum_{s \in \mathbb{N}} \left( 1 - \prod_{i=1}^{N_1} \left( \sum_{o_1, o_2 \in \mathbb{Z} \mid o_1 + o_2 = s} \left( \Pr(L_{i,1} = o_1) \times \prod_{c_{k,2} \in P_{i,1}} F_{i,2}(X_{i,2} + o_2) \right) \right) \right) \\ & + \sum_{i=1}^{N_1} H_{i,1} \times \left( \sum_{s \in \mathbb{N}} \left( 1 - \prod_{c_{k,2} \in S_{i,1}} F_{k,2}(X_{k,2} + s) \right) \right) \end{aligned}$$

$$\begin{aligned}
& - \sum_{i=1}^{N_1} H_{i,1} \times \left( \sum_{s \in \mathbb{N}} \left( 1 - \prod_{c_{k,2} \in S_{i,1}} (1 - F_{k,2}(X_{k,2} - s - 1)) \right) \right) \\
& - \sum_{i=1}^{N_1} \left( \sum_{c_{k,2} \in P_{i,1}} h_{k,2} \{E(L_{k,2}) - X_{k,2}\} \right) - \sum_{i=1}^{N_1} h_{i,1} E(L_{i,1})
\end{aligned} \tag{5.1}$$

où,

$$2 \leq X_{k,2} \leq U_{k,2}, \text{ pour } k=1, 2, \dots, N_2 \tag{5.2}$$

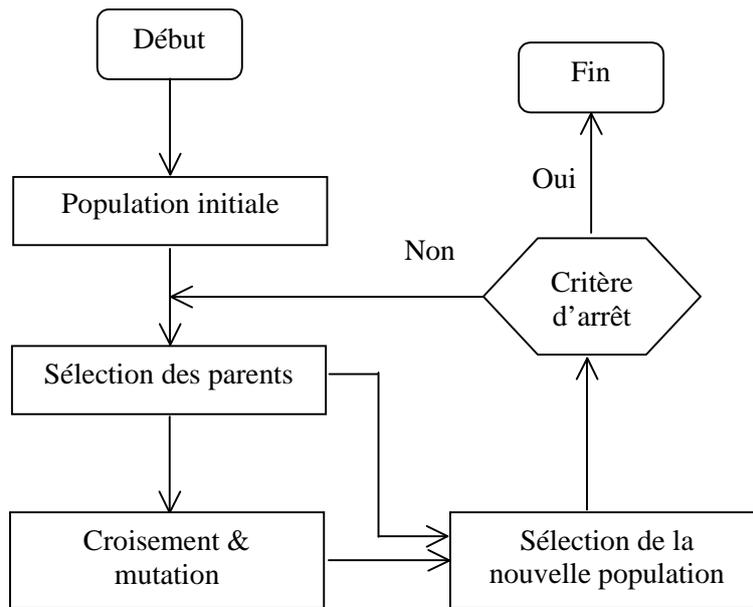
$$X = (X_{1,2}, \dots, X_{k,2}, \dots, X_{N_2,2})$$

$$H_{i,1} = -h_{i,1} + \sum_{c_{k,2} \in P_{i,1}} h_{k,2}$$

Le problème considéré correspond à la minimisation d'une fonction objectif non linéaire à variables de décision entières, voir (5.1)-(5.2). La fonction objectif  $EC(X)$  est calculée en un temps polynomial en  $O(N_2 * U)$  (avec  $U: \max (U_{k,2}), k=1,2,\dots,N_2$ ). Mais la question de savoir si le problème est lui-même solvable en un temps polynomial ou s'il est NP-difficile est ouverte.

Etant donnée l'explosion combinatoire, l'énumération de toutes les solutions devient vite prohibitive quand la taille du problème augmente. Comme nous l'avons vu dans le chapitre précédent, la PSE utilisée permet de résoudre des problèmes de petite et moyenne tailles (la taille maximum du problème est de 60 variables avec  $U=10$ ) avec un temps CPU raisonnable. Donc, dans le but de résoudre les problèmes de grande taille dans un temps raisonnable, un algorithme génétique est développé dans ce chapitre.

Les principes fondamentaux des algorithmes génétiques sont décrits dans la figure 5.1. L'algorithme génétique commence par la création d'un ensemble des solutions de départ appelé souvent population initiale. Ensuite, il utilise des opérateurs de reproduction (croisement et mutation) ainsi que des opérateurs de sélection pour améliorer la qualité de la solution. Ces opérateurs sont répétés à chaque génération jusqu'à une condition d'arrêt (temps d'exécution, nombre des générations, etc.).



**Figure 5.1 :** Les principes fondamentaux d'un algorithme génétique

Ces principes sont décrits plus en détail dans l'algorithme proposé dans notre étude.

L'algorithme considéré dans notre étude prend les principes de base d'un algorithme génétique. Toutefois, en raison de son comportement stochastique, un algorithme génétique peut souffrir d'une convergence lente, et de l'instabilité des résultats. Donc, nous avons pris en considération cet aspect en proposant des procédures de recherche locale pour accélérer la convergence de l'algorithme. Ainsi, l'algorithme génétique proposé est classé dans la classe des algorithmes « Memetic » selon Moscato *et al.* (2000).

L'algorithme proposé ici est présenté dans la figure 5.2. Les différentes étapes qui le composent sont présentées en détail par la suite.

### Algorithme génétique

```
Fonction Sous-ensemble_Meilleur ( $A, n$ )  
    retourne  $S \subseteq A, |S| = n$  et  $\forall s \in S, \nexists s' \in A \setminus S, \text{Fitness}(s') < \text{Fitness}(s)$   
Fin Fonction  
Ensemble_Population  $\leftarrow$  Population_Initiale ( $N$ )  
Ensemble_Optimum_Local  $\leftarrow \emptyset$   
Pour  $i$  dans  $1, \dots, \text{Max\_génération}$   
    // Sélection de reproduction //  
    Ensemble_Parents  $\leftarrow$  Sous-ensemble_Meilleur (Ensemble_Population,  $N/2$ )  
    //Opérateurs de reproduction //  
    Ensemble_Fils  $\leftarrow$  Croisement (Ensemble_Parents)  
    Ensemble_Fils  $\leftarrow$  Mutation (Ensemble_Fils)  
    // Recherche Locale //  
    Meilleure_Solution  $\leftarrow$  Sous-ensemble_Meilleur (Ensemble_Parents  $\cup$   
        Ensemble_Fils  $\setminus$  Ensemble_Optimum_Local,  $1$ )  
    Meilleure_Dans_Voisinage (Meilleure_Solution, Solution_RL)  
    Si Solution_RL = Meilleure_Solution  
        Ensemble_Optimum_Local  $\leftarrow$  Ensemble_Optimum_Local  $\cup$  {  
            Solution_RL }  
    Fin Si  
    // Sélection de Remplacement //  
    Ensemble_Population  $\leftarrow$  Sous-ensemble_Meilleur (Ensemble_Parents  
         $\cup$  Ensemble_Fils  $\cup$  { Solution_RL  $\setminus$  {  
            Meilleure_Solution } },  $N$ )  
Fin Pour  
Fin Algorithme
```

**Figure 5.2 :** Algorithme génétique pour le problème PP1

Nous commençons par présenter le codage choisi pour les solutions ainsi que la fonction Fitness utilisée dans notre algorithme. Ensuite, nous expliquerons notre choix de la population initiale. Par la suite, la procédure de sélection des individus ‘‘solutions’’ qui vont subir les opérations de reproduction et la procédure de sélection des individus pour la génération suivante de l’algorithme sont décrites en détail. Dans l’étape suivante, nous présentons les opérateurs de reproduction (croisement et mutation). Enfin, nous présentons la procédure de recherche locale.

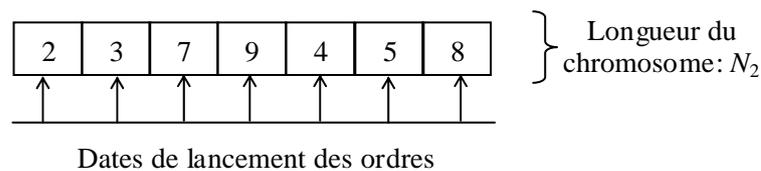
## 5.2.1 Codage des solutions et fonction Fitness

### 5.2.1.1 Codage des solutions

La première étape d'un algorithme génétique est de représenter convenablement les variables de décision du problème à traiter. Un des facteurs les plus importants, si ce n'est le plus important, est la façon dont sont codées les solutions (les chromosomes), c'est-à-dire les structures de données qui coderont les gènes. Le choix du codage dépend de la spécificité du problème et conditionne fortement l'efficacité de l'algorithme. Une représentation de la solution devrait être complète, ce qui signifie que toutes les solutions possibles au problème peuvent être codées à l'aide de cette représentation.

Dans notre cas d'étude, les variables de décision du problème considérées sont les dates de lancement des ordres (variables de décision entières), donc chaque chromosome peut être codé avec un tableau de nombres entiers. Chaque gène d'un chromosome représente une date de lancement d'ordre. La longueur du chromosome est égale à  $N_2$  (nombre de composants de niveau 2). Par conséquent, ce codage garantit que toutes les solutions du problème puissent être représentées.

La figure 5.3 présente un exemple des solutions pour le problème avec  $N_2 = 7$ .



**Figure 5.3 :** Représentation de chromosome

Chaque chromosome de l'algorithme génétique est évalué à travers une fonction appelée fonction Fitness. Dans la section suivante nous définissons cette fonction plus en détail.

### 5.2.1.2 Fonction Fitness

Il faudra définir pour l'ensemble des chromosomes une fonction d'évaluation appelée généralement une fonction Fitness. Cette fonction doit être capable d'interpréter les données contenues dans un chromosome (les gènes) et de lui attribuer une valeur numérique. A partir de ces valeurs numériques, une sélection des chromosomes est faite pour la reproduction par les opérateurs de croisement et de mutation. La solution dont les gènes ne forment pas une bonne solution se verra attribuer une mauvaise valeur de la fonction Fitness et aura une probabilité forte d'être éliminé par le processus de sélection. Cependant, il peut être intéressant de conserver certaines solutions de mauvaise valeur de

fonction Fitness car elles peuvent permettre de générer d'autres solutions, par les opérateurs de reproduction (croisement et mutation), admissibles et de bonne qualité.

Dans notre cas, l'espérance mathématique de coût total  $EC(X)$ , voir équation (5.1), est utilisée pour évaluer la fonction Fitness de chaque individu de la population. Ainsi, chaque chromosome a une évaluation unique qui sera décisive pour la phase de sélection.

### 5.2.2 Génération de la population initiale

La population initiale est générée le plus souvent aléatoirement en faisant des tirages uniformes dans chacun des domaines associés aux variables de décision, tout en veillant à ce que les individus produits respectent les contraintes du problème. Le choix de la population initiale d'individus conditionne fortement la rapidité de l'algorithme génétique. Une connaissance de solutions de bonne qualité permettrait à l'algorithme de converger plus rapidement vers des solutions de bonne qualité. Ainsi, une voie prometteuse pour améliorer la population initiale est d'intégrer certains chromosomes générés par des heuristiques. La taille de la population initiale ( $N$ ) qui est un paramètre de l'algorithme génétique doit être choisi convenablement. Si cette taille est trop grande, l'algorithme risque de consommer beaucoup de temps à générer cette population. Dans le cas contraire, si elle trop petite l'algorithme risque de ne pas avoir assez de variété dans la population. Les chromosomes générés doivent être sans doublons afin de donner une bonne diversité à la population de départ.

Dans l'algorithme proposé, la population initiale se compose des  $N$  solutions en veillant à ce que toutes les solutions soient différentes. En effet, un test est fait à la fin de la procédure de création de la population initiale qui permet d'éliminer les doublons et les remplacer par d'autres solutions générées d'une manière aléatoires. Les solutions qui constituent la population initiale sont les suivantes :

- La solution utilisant la somme des espérances mathématiques des délais d'approvisionnement :  $(E(L_{k,2}) + E(L_{i,1}))$  pour chaque  $X_{k,2}$ ,  $i= 1, 2, \dots, N_1$  et  $c_{k,2} \in P_{i,1}$ . La solution choisie, notée par  $\bar{X}_{k,2}$ , est la suivante :

$$\bar{X}_{k,2} = \begin{cases} \lfloor E(L_{k,2}) + E(L_{i,1}) \rfloor & \text{si } (E(L_{k,2}) + E(L_{i,1})) - \lfloor E(L_{k,2}) + E(L_{i,1}) \rfloor < 0.5 \\ \lceil E(L_{k,2}) + E(L_{i,1}) \rceil & \text{sinon} \end{cases} \quad (5.3)$$

- La solution où chaque variable  $X_{i,2}$ ,  $i=1, 2, \dots, N_2$  correspond à la somme des délais d'approvisionnement les plus probables à chaque niveau :

$$X_{i,2} = x_{i,1} + x_{k,2} \quad (5.4)$$

où,

$$c_{k,2} \in P_{i,1}$$

$$\Pr(L_{i,1} = x_{i,1}) = \max_{r \in [1, \dots, u_{i,1}]} (\Pr(L_{i,1} = r)) \quad (5.5)$$

$$\Pr(L_{k,2} = x_{k,2}) = \max_{r \in [1, \dots, u_{k,2}]} (\Pr(L_{k,2} = r)) \quad (5.6)$$

- Les solutions qui assurent un certain niveau de service : les solutions où chaque variable  $X_{i,2}$ ,  $i=1,2, \dots, N_2$ , correspond à un niveau de la fonction cumulée de répartition du délai d'approvisionnement. Les niveaux suivants sont utilisés : 0,75; 0,80; 0,85; 0,90 et 0,95 :

$$X_{i,2} = \text{Min} \{ S_i \mid \Pr(L_{i,1} + L_{k,2} \leq S_i) \geq SL \} \quad (5.7)$$

Avec :

$$SL \in \{0,75; 0,80; 0,85; 0,90; 0,95\} \quad (5.8)$$

$$c_{k,2} \in P_{i,1}$$

- Les solutions avec les mêmes valeurs pour toutes les variables, pour couvrir l'espace de recherche entier  $k \in [2, \dots, U]$  :

$$X_{i,2} = \text{Min}(k, U_{i,2}), \quad \forall i \in [1, \dots, N_2] \quad (5.9)$$

Il y a donc  $(6 + U)$  solutions générés par les équations (5.3)-(5.9). Dans le cas où ce nombre dépasserait la taille de la population initiale  $N$  fixe, on élimine les solutions avec la valeur de la fonction fitness la moins bonne afin de garder la taille de la population à  $N$ . Dans le cas où il manque encore des solutions, le reste de la population est généré d'une manière aléatoire selon une distribution uniforme comme suit :

$$X_{i,2} = \text{rand}(2, U_{i,2}) \text{ pour } i \in [1, 2, \dots, N_2] \quad (5.10)$$

où

$\text{rand}(2, U_{i,2})$  est une valeur aléatoire discrète uniformément distribuée entre 2 et  $U_{i,2}$ .

### 5.2.3 Sélection

Une fois la population initiale est créée et évaluée, certains individus sont sélectionnés pour participer à la création de la nouvelle génération. Les individus peuvent être sélectionnés au moyen de plusieurs techniques que la littérature peut fournir dont nous citons : la sélection aléatoire, la sélection par tournois, la sélection  $N/2$ -élitisme, la sélection par roulette et la sélection par rang. Pour plus d'informations sur les méthodes de sélection, nous envoyons à Bäck (1994), Goldberg et Deb (1991).

Notre algorithme génétique considère deux phases de sélection :

1. La sélection de reproduction, c'est la sélection des parents défini dans la figure 5.1, qui détermine les individus qui vont subir les opérations de reproduction (croisement et mutation). La sélection est  $N/2$  élitisme, elle consiste à sélectionner les  $N/2$  meilleures solutions dans la population courante, en utilisant la fonction « Sous-ensemble\_Meilleur » (défini dans l'algorithme génétique, voir la figure 5.2).
2. La sélection de remplacement, la sélection de la nouvelle population défini dans la figure 5.1, concerne l'évolution de la population d'une génération à une autre. La taille de la population est fixée à  $N$ . Donc seulement  $N$  meilleures solutions sont gardées pour la population suivante. Elles sont choisies parmi les  $N+N/2$  solutions ( $N$  parents de la population courante (Ensemble\_Parents) et  $N/2$  enfants obtenus par la reproduction (Ensemble\_fil)). Cette sélection est donc élitiste. Il est à noter que les individus doublons sont supprimés à cette étape afin de maintenir une diversité suffisante de la population.

#### 5.2.4 Opérateur de croisement

Le croisement a pour but d'intensifier la recherche de meilleures solutions. Classiquement, les croisements sont envisagés avec deux parents et génèrent deux enfants. Cela consiste à échanger les gènes des parents afin de donner des enfants qui portent des propriétés combinées. La création d'un couple d'enfants à partir d'un couple des parents est effectuée avec une probabilité de croisement. Il existe différents processus de croisement pour créer deux enfants à partir d'un couple de parents. Parmi ces processus, nous pouvons citer le croisement en  $n$  ( $n \in [1, \dots, N_2 - 1]$ ) points. Ces  $n$  points représentent les  $n$  points de coupure de chaque parent. Chaque parent est composé donc de chaînes de gènes. Pour générer un couple d'enfant, il suffit de choisir une chaîne de gènes d'un parent et une chaîne de gènes d'un autre parent.

Dans notre algorithme génétique, vu la structure des solutions du problème, l'opérateur de croisement classique à un seul point est mis en place pour combiner les solutions de l'ensemble des parents. L'ensemble des  $N/2$  parents sélectionnés est partitionné de manière aléatoire en  $N/4$  couples. Le point de croisement est aussi choisi de manière aléatoire à chaque génération de l'algorithme dans l'intervalle  $[1, \dots, N_2-1]$ . Chaque couple subit une opération de croisement avec une probabilité de croisement. Les couples qui ne subissent pas une opération de croisement sont maintenus.

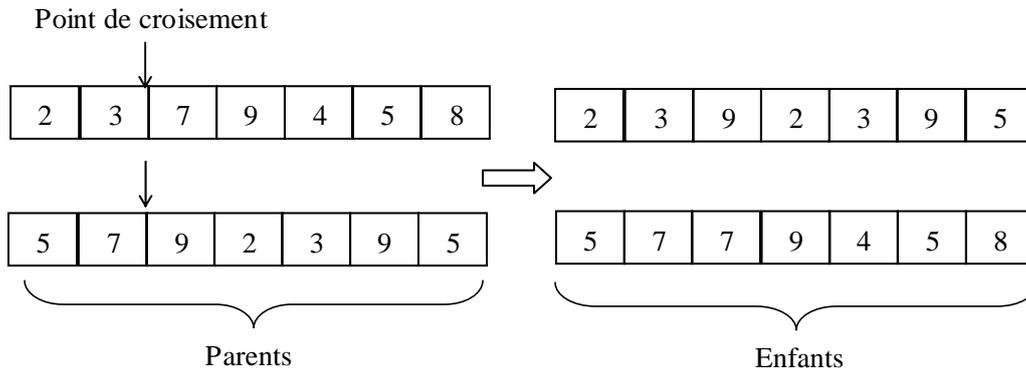
Chaque couple de parents  $X^{p1} = (X_{1,2}^{p1}, \dots, X_{i,2}^{p1}, \dots, X_{N_2,2}^{p1})$  et  $X^{p2} = (X_{1,2}^{p2}, \dots, X_{i,2}^{p2}, \dots, X_{N_2,2}^{p2})$  subit l'opération de croisement à un point. Les deux enfants  $X^{o1} = (X_{1,2}^{o1}, \dots, X_{i,2}^{o1}, \dots, X_{N_2,2}^{o1})$  et  $X^{o2} = (X_{1,2}^{o2}, \dots, X_{i,2}^{o2}, \dots, X_{N_2,2}^{o2})$  sont obtenus comme suit :

$$\begin{cases} X_{i,2}^{ok} = X_{i,2}^{pk} & i \in [1, 2, \dots, j] \text{ et } k \in \{1, 2\} \\ X_{i,2}^{o1} = X_{i,2}^{p2} & i \in [j+1, \dots, N_2] \\ X_{i,2}^{o2} = X_{i,2}^{p1} & i \in [j+1, \dots, N_2] \end{cases} \quad (5.11)$$

Avec :

$j$  est le point de croisement.

Dans la figure 5.4, un exemple d'opérateur de croisement, où la taille du chromosome ( $N_2$ ) est égale à 7 et le croisement est au point 2, est montré.



**Figure 5.4 :** Opérateur de croisement

### 5.2.5 Opérateur de mutation

L'objectif majeur d'un opérateur de mutation est d'apporter à l'algorithme génétique une diversification nécessaire pour une exploration efficace de l'espace de recherche. Sa mise en œuvre doit permettre à l'algorithme d'atteindre la plupart des sous-espaces de solution réalisables. En effet, la mutation joue le rôle d'un bruit et empêche l'évolution de se figer. Les propriétés de convergence d'un algorithme génétique sont fortement dépendantes de l'opérateur de mutation. L'opérateur de mutation est utilisé avec une probabilité, appelée probabilité de mutation, généralement choisie faible.

Dans notre étude, chaque solution  $X = (X_{1,2}, \dots, X_{i,2}, \dots, X_{N_2,2})$  peut subir une modification aléatoire, voir l'expression 5.12. La mutation concerne uniquement un gène  $X_{i,2}$  du chromosome. Ce gène est sélectionné aléatoirement à chaque génération par une simple distribution uniforme.

$$\begin{cases} X_{i,2}^{new} = X_{i,2}^{old} + \text{rand}(-2, 2) \\ X_{j,2}^{new} = X_{j,2}^{old} \quad \forall j \in \{1, \dots, N_2\}, j \neq i \end{cases} \quad (5.12)$$

Où  $\text{rand}(-2, 2)$  est une variable aléatoire discrète uniformément distribuée comprise entre -2 et 2.

Comme chaque variable de décision de problème d'optimisation est bornée par les limites supérieures et inférieures, alors :

Si  $X_{i,2}^{new} > U_{i,2}$  alors  $X_{i,2}^{new} = U_{i,2}$  et si  $X_{i,2}^{new} < 2$  alors  $X_{i,2}^{new} = 2$ .

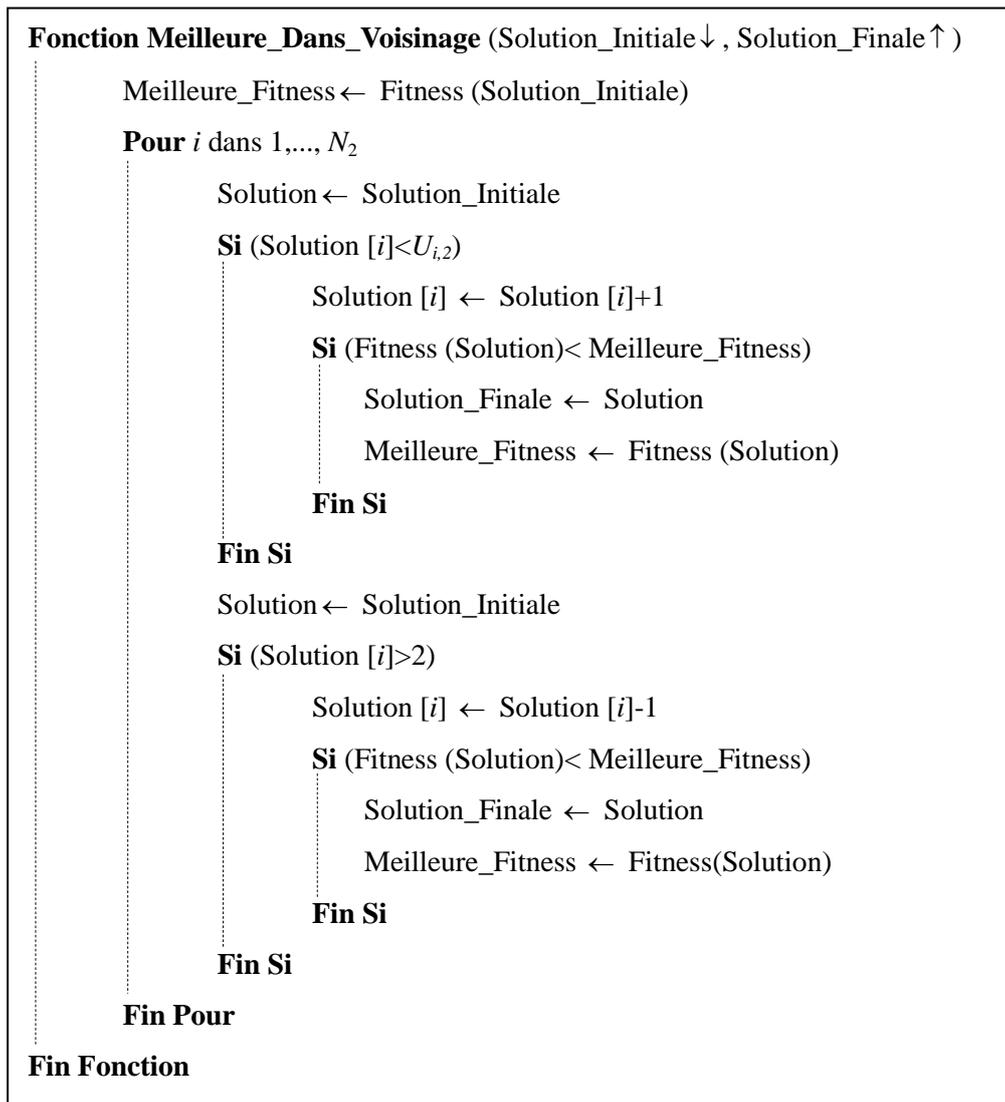
### 5.2.6 Recherche locale

En plus des opérateurs de reproduction classiques d'un algorithme génétique qui sont décrits précédemment, il est souvent utile d'intégrer une recherche locale afin d'accélérer la convergence de l'algorithme ( Cheng *et al.*, 1997 ; França *et al.*, 2001 ; Dolgui *et al.*, 2007 ; Caumont *et al.*, 2007).

Pour notre problème, la recherche de voisinage suivante a été considérée : la procédure commence par une solution, toutes les solutions avec une variation d'une unité d'un gène sont explorées. Ainsi, l'ensemble de voisinage peut contenir jusqu'à  $2 \times N_2$  solutions. La meilleure solution parmi cet ensemble de voisinage est alors choisie (voir la figure 5.5). Il est à noter que la solution obtenue n'est pas nécessairement un optimum local (car nous appliquons une recherche de voisinage une seule fois à chaque génération). Une nouvelle recherche locale est appliquée à cette solution à la génération suivante si elle est toujours la meilleure solution dans la population.

Quand la meilleure solution trouvée par la procédure de recherche de voisinage est la solution initiale, cela signifie que cette solution est un optimum local et, par conséquent, elle est archivée dans un ensemble des optimum locaux appelé « Ensemble\_Optimum\_Local » (défini dans l'algorithme génétique, voir la figure 5.2).

Compte tenu du temps de calcul non négligeable utilisé pour évaluer la fonction Fitness d'une solution, l'application d'une telle recherche locale sur chaque individu de la population consommerait beaucoup de temps. Par conséquent, dans la présente étude, la recherche locale n'est appliquée, à chaque itération, que sur la meilleure solution qui n'est pas un minimum local (i.e., qui n'est pas dans Ensemble\_Optimum\_Local).



**Figure 5.5** : Procédure pour obtenir le meilleur voisinage de la solution

Dans la section suivante, l'algorithme proposé est évalué. Une série de tests est faite, avec différents paramètres, afin d'examiner la robustesse de l'algorithme sur différents scénarios de la chaîne d'approvisionnements. L'effet bénéfique de la recherche locale sur la performance de l'algorithme génétique, en terme de qualité de la solution, sur la convergence et le temps de calcul, est également présenté dans la section suivante.

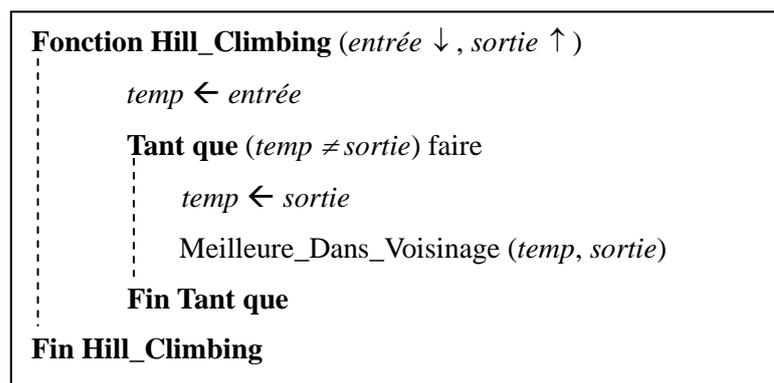
### 5.3 Résultats expérimentaux

Les tests consistent à exécuter l'algorithme sur 120 instances générées d'une manière aléatoire. Pour ces instances, le nombre de composants au niveau 2 ( $N_2$ ) varie dans l'intervalle [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120]. Pour chaque nombre de composants ( $N_2$ ), 10 instances différentes sont

générées (i.e., nous générons 10 instances différentes pour chaque taille de problème). Les données de chaque instance sont : la fonction de distribution des délais d'approvisionnements des composants, les coûts unitaires de stockage des composants, le coût unitaire de rupture en produit fini et le nombre des composants au niveau 1 nécessaires pour l'assemblage de produit fini et le nombre des composants de niveau 2 nécessaires pour l'assemblage de chaque type des composants du niveau 1.

A notre connaissance, aucune autre heuristique n'a été proposée pour ce problème, donc, la performance de l'algorithme génétique a été comparée avec les trois heuristiques simples suivantes :

- L'heuristique qui utilise les valeurs moyennes des délais d'approvisionnement (espérance mathématiques) notée par "Heuristique".
- Un algorithme de recherche locale de type « Hill-Climbing » (voir la figure 5.6) qui débute de la solution moyenne et utilise ensuite une recherche locale itérative présentée dans la section 5.2.6. Il est noté par "HC".
- La borne supérieure de la PSE du chapitre précédent notée par "BS".



**Figure 5.6 :** La procedure de Hill-Climbing

### 5.3.1 Conditions expérimentales

L'algorithme génétique décrit dans la section 5.2 a été implémenté en C++. Les tests ont été exécutés sur SUN UltraSPARC IIIi avec une fréquence de 1593 Mhz CPU et 16 GB de mémoire.

Après plusieurs tests préliminaires, les valeurs suivantes ont été choisies empiriquement comme paramètres de notre algorithme génétique : La taille de la population  $N$  est égale à 60 chromosomes. La probabilité de croisement est égale à 0,80 et la probabilité de mutation est égale à 0,15. Le nombre de générations « Max\_génération » (i.e., le test arrêt est donné par Max\_génération) est égal à 1000. Considérant le caractère stochastique de l'algorithme génétique, 100 lancements (un lancement ou run=une exécution de l'algorithme) indépendants ont été effectués pour chaque instance. En plus, les

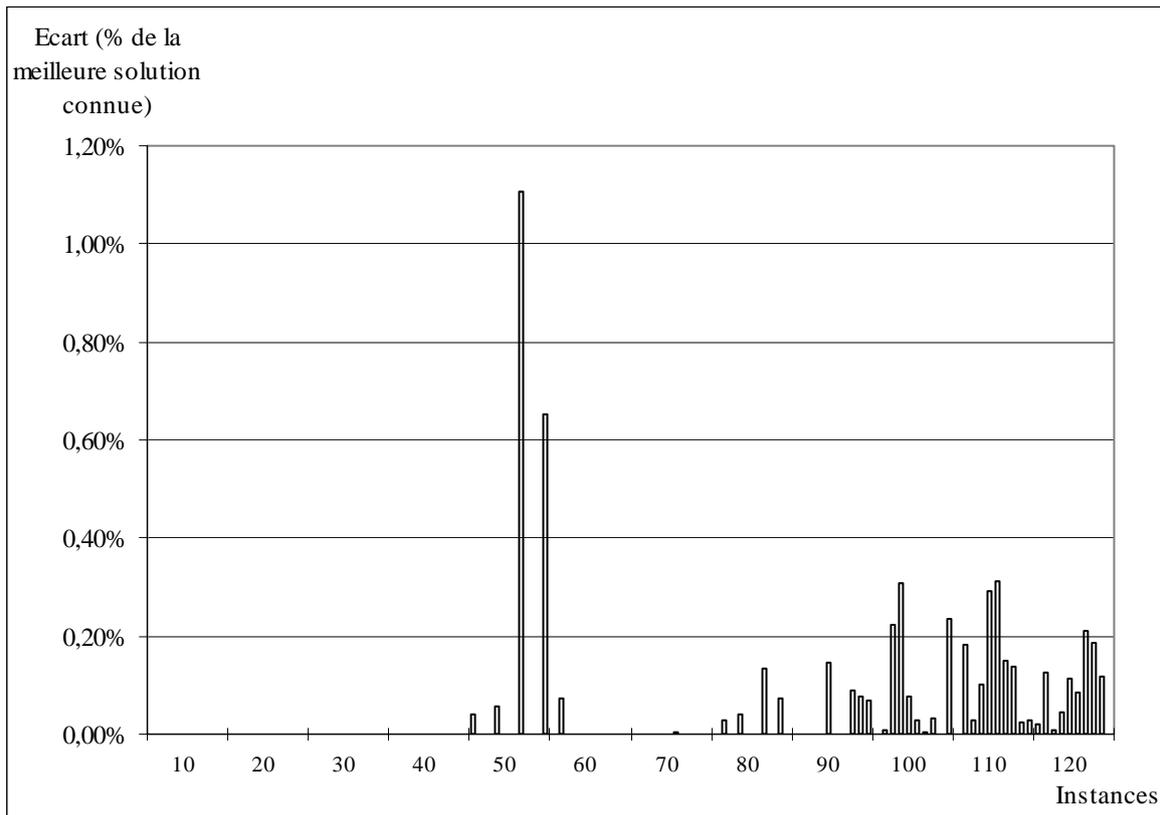
résultats concernent 10 instances à chaque fois (pour chaque type, ç.-à.-d. 10 instances différentes avec le même nombre de composants), et sont exprimés en pourcentage de la meilleure solution connue. La meilleure solution connue pour chaque test correspond à la solution optimale donnée par la PSE présentée dans le chapitre précédent si la PSE peut résoudre ce problème. Dans le cas où la PSE n'arriverait pas à résoudre le problème, ce qui est généralement le cas pour les problèmes de grande taille, la meilleure solution connue correspond à la solution obtenue parmi les 100 runs, parmi les tests préliminaires effectués pour paramétrer l'algorithme génétique et aussi parmi les solutions obtenues avec d'autres heuristiques (section 5.3.2).

Il est à noter que pour les instances où le nombre des composants au niveau 2 varie dans l'intervalle [10,...,60], la solution optimale est donnée par la PSE.

### **5.3.2 Résultats expérimentaux de l'algorithme génétique**

Tout d'abord, nous examinons les résultats des tests de l'algorithme génétique qui ont été faits sans utiliser la procédure de recherche locale.

Une observation intéressante est que les résultats obtenus sont stables. En fait, pour 65% des instances, la valeur obtenue pour le meilleur run est égale à celle obtenue pour le pire run. Si les runs extrêmes ne sont pas considérés, c'est-à-dire 10% des runs ne sont pas considérés (les meilleurs 5% et les pires 5%), alors ce pourcentage augmente à 88,3%. Si l'on examine l'écart entre les meilleurs et les pires runs pour chaque instance (voir la figure 5.7), cet écart semble également très faible : 0,05% en moyenne et 1,11% dans le pire des cas (c'est-à-dire l'instance avec l'écart maximum entre les 120 instances considérées).

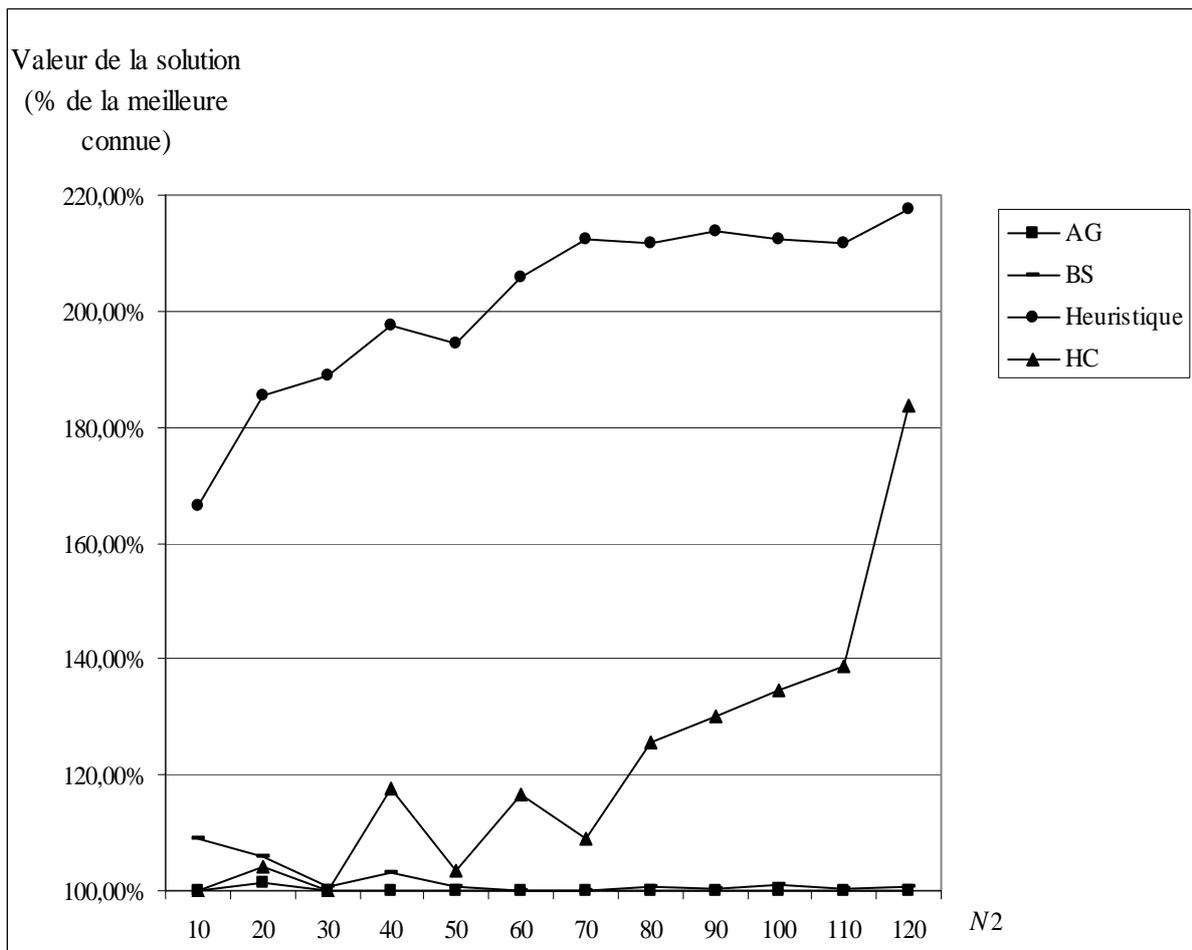


**Figure 5.7 :** L'écart entre la meilleure et le pire run de chaque instance

La figure 5.8 présente une étude comparative entre l'algorithme génétique (AG) proposé et les trois heuristiques présentées précédemment (Heuristique, HC et BS) en termes de qualité des résultats obtenus. Sur cette figure nous présentons, pour chacune des heuristiques et pour chaque taille de problème l'écart entre la meilleure solution connue (en moyenne sur les instances de même taille et tous les runs correspondants) et la solution obtenue par l'heuristique. La méthode "Heuristique" semble incapable d'obtenir de bonnes solutions. Il est à noter qu'ici nous avons un écart énorme (de 60% à 120%) par rapport à la meilleure solution connue. Pourtant cette heuristique est souvent utilisée dans la pratique. Ceci justifie l'utilisation de méthodes d'optimisation que nous développons, parce que le bénéfice obtenu peut être significatif.

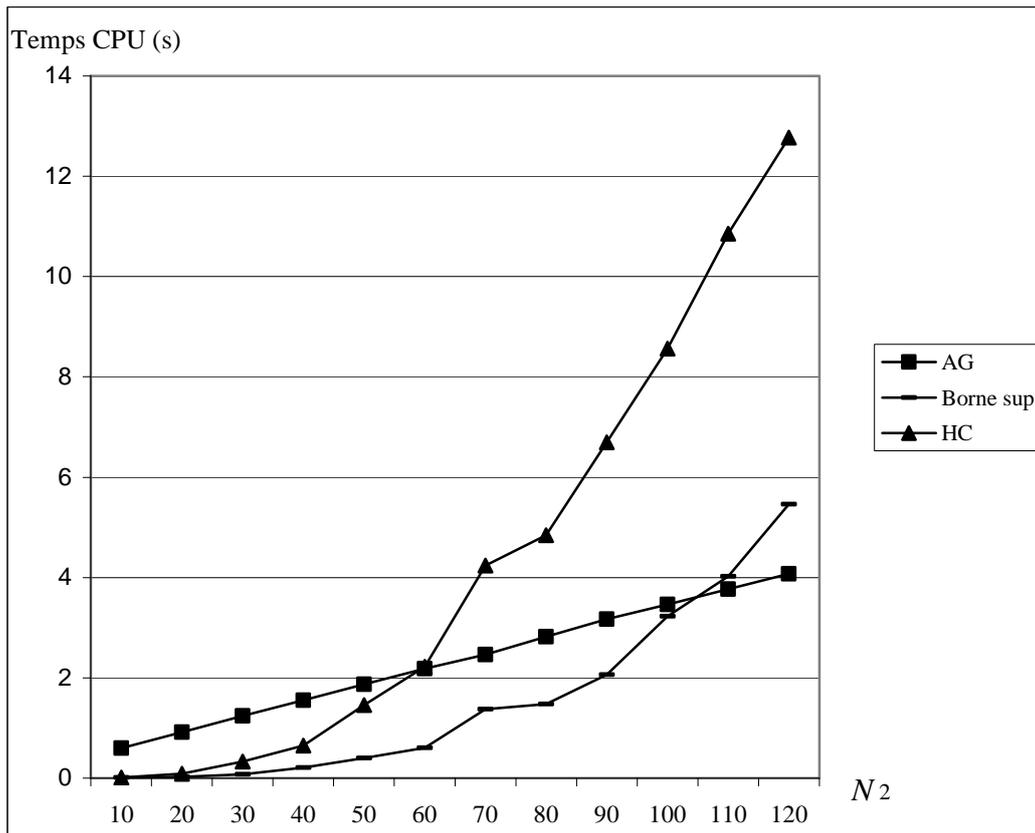
L'ajout de la procédure Hill-Climbing les performances sont meilleures pour les instances de petite taille, mais elles baissent rapidement quand la taille de problèmes augmente.

Pour l'ensemble des instances, l'algorithme génétique fournit des solutions en moyenne à 0,13% de la meilleure solution connue, alors que la Borne supérieure se situe à 1,9%. Selon la taille de problème, cet écart varie de 0% à 1,53% pour l'algorithme génétique, et de 0,16% à 8,82% pour l'heuristique calculant la borne supérieure. En outre, quelle que soit la taille de problème, l'écart de l'algorithme génétique est toujours inférieur à celui de l'algorithme calculant la borne supérieure.



**Figure 5.8 :** La performance moyenne en fonction de la taille des problèmes

Dans la figure 5.9, les temps de calcul "CPU" moyen de l'algorithme génétique, de l'heuristique calculant la borne supérieure et de l'heuristique Hill-Climbing sont présentés. Le temps CPU de l'Heuristique est évidemment très faible et n'est pas représenté dans cette figure. Pour les instances considérées, le temps CPU est acceptable pour l'utilisation pratique sur de vrais problèmes de taille importante. Le temps CPU de l'algorithme génétique augmente linéairement. Toutefois, pour les deux autres heuristiques le temps CPU commence avec des valeurs faibles et ensuite augmente rapidement pour les instances de taille importante. Pour les problèmes avec plus de 120 composants au niveau 2, l'algorithme génétique devrait devenir nettement plus rapide que les autres heuristiques. Ainsi, compte tenu de la qualité de la solution et du temps CPU, l'algorithme génétique dépasse clairement les deux autres heuristiques.



**Figure 5.9 :** Le temps de calcul “CPU ” en fonction de la taille de problème

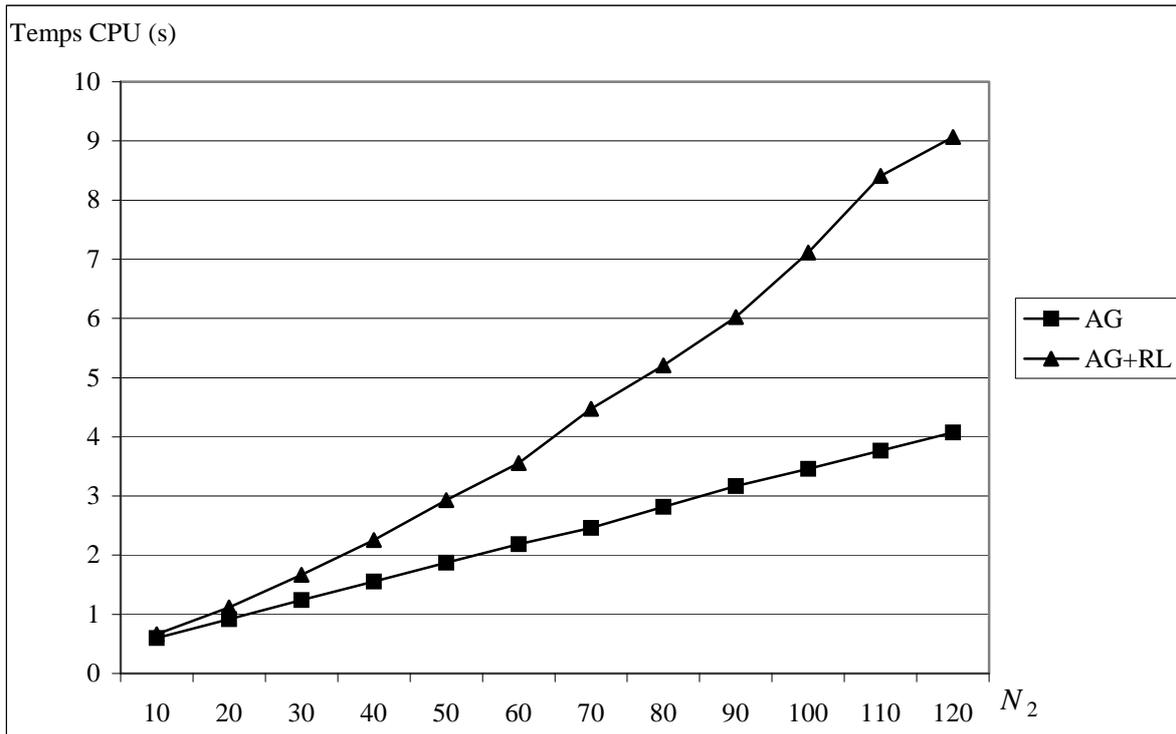
### 5.4.2 Impact de la recherche locale

Maintenant, une étude comparative de l'algorithme génétique avec recherche locale (AG + RL) et sans recherche locale (AG) sera présentée afin de déterminer l'impact de la recherche locale.

Dans le tableau 5.1, les performances moyennes de l'algorithme génétique (AG + RL) et (AG) sont présentées. Nous remarquons que les performances des deux algorithmes sont très similaires et très bonnes. Globalement, l'algorithme génétique avec une recherche locale fournit des solutions avec un écart en moyenne de 0,02% par rapport à la meilleure solution connue et de 0,13% pour l'algorithme génétique sans recherche locale. D'autre part, l'ajout de la phase recherche locale a augmenté le temps CPU consommé par l'algorithme génétique (voir la figure 5.10)

**Table 5.1 :** Performances moyennes des algorithmes génétiques (AG) et (AG + RL)

N <sub>2</sub>	10	20	30	40	50	60	70	80	90	100	110	120
AG	100%	101,53%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
AG+ RL	100%	100%	100%	100%	100,05%	100%	100%	100%	100%	100%	100%	100,17%

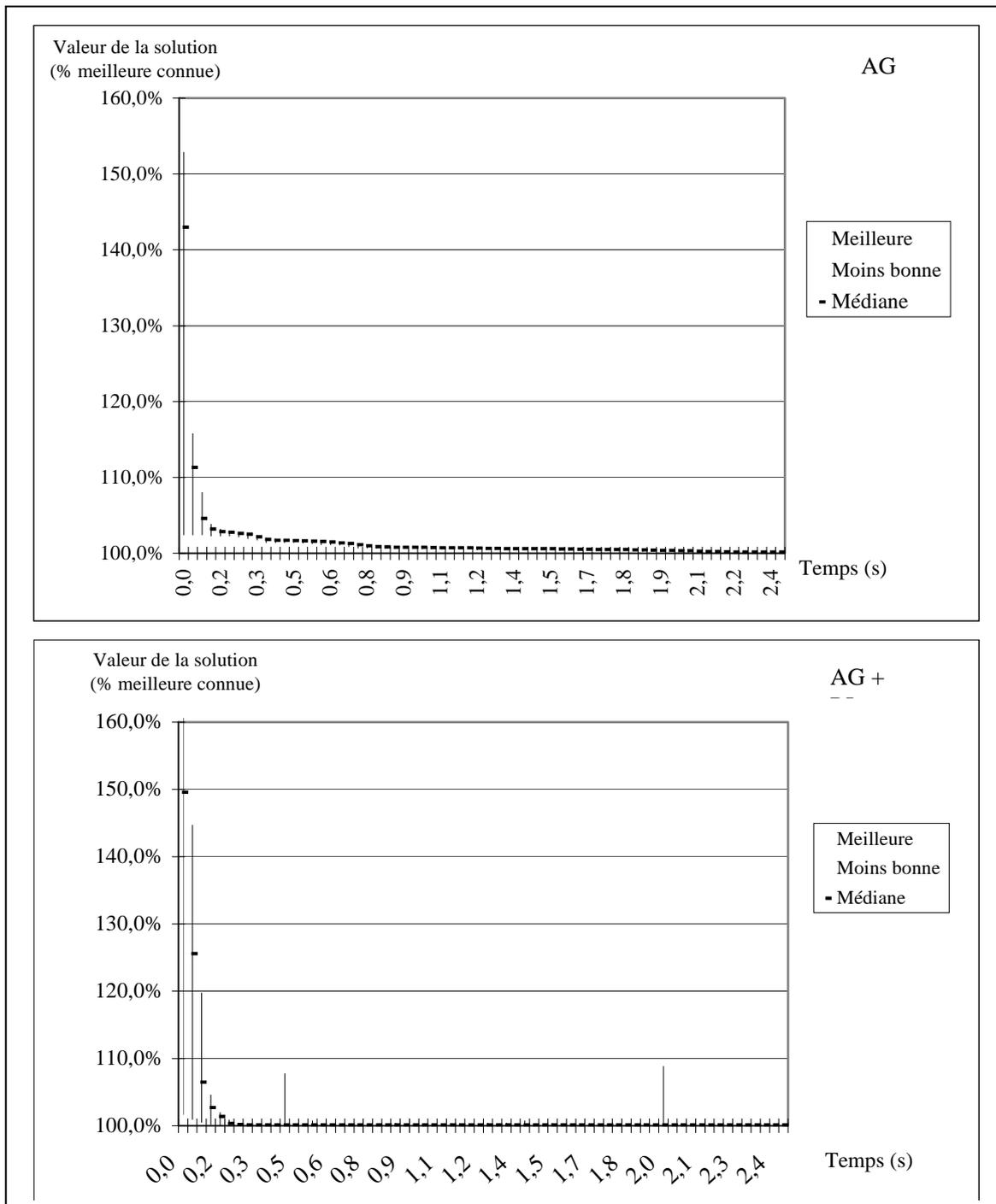


**Figure 5.10 :** Le temps CPU moyen des algorithmes AG et AG+RL

Cependant l'impact principal de la recherche locale concerne la convergence de l'algorithme génétique. Nous présentons ainsi dans le tableau 5.2 le temps moyen nécessaire pour que les deux algorithmes génétiques AG et AG+RL trouvent leur solution finale. Nous pouvons visualiser que l'ajout de la recherche locale a vraiment accéléré la convergence de l'algorithme. En plus, le ratio de temps ne diminue pas pour les instances de grande taille. Cette performance est illustrée dans la figure 5.11, en regardant l'évolution de la répartition de la population des deux algorithmes sur une instance spécifique. Il est à noter que l'élimination des doublons au sein de la population peut forcer l'apparition de mauvaises solutions à l'AG + RL, mais de telles solutions disparaissent rapidement.

**Table 5.2 :** Le temps moyen de convergence des AG et AG+RL (en secondes)

N <sub>2</sub>	10	20	30	40	50	60	70	80	90	100	110	120
AG	0,012	0,062	0,151	0,289	0,485	0,589	0,922	1,144	1,265	1,617	2,243	2,163
AG +RL	0,002	0,010	0,083	0,057	0,109	0,100	0,259	0,158	0,372	0,248	0,423	0,615
Ratio : $\frac{AG}{AG + RL}$	5,196	5,967	1,833	5,072	4,455	5,901	3,561	7,256	3,404	6,529	5,303	3,516



**Figure 5.11 :** Exemple de convergence de la population pour un « run » d’une instance de taille 120

## 5.5 Conclusion

Dans ce chapitre, nous avons étudié l’optimisation de la gestion des stocks et de la planification d’approvisionnements pour le cas des systèmes d’assemblage à deux niveaux lorsque les délais d’approvisionnements en composants sont aléatoires. Le critère étudié est la somme de l’espérance

mathématique des coûts de stockage des composants et l'espérance mathématique du coût de rupture en produit fini.

Un algorithme génétique a été proposé pour réduire le coût total moyen. Cet algorithme a été testé sur un grand nombre d'instances et une comparaison avec plusieurs heuristiques a été réalisée. L'effet d'une recherche locale sur les performances de l'algorithme génétique en termes de qualité de solution, de la convergence et de temps de calcul a aussi été étudié et présenté dans ce chapitre. Les résultats numériques montrent l'efficacité de notre algorithme ainsi que la capacité de la recherche locale d'accélérer la convergence de l'algorithme génétique.

# Conclusions générales

Face à une intensification de la concurrence, la gestion des systèmes de production a beaucoup changé ces dernières années. Les industriels sont toujours sollicités à bien gérer leurs entreprises et à prendre les bonnes décisions au bon moment afin de satisfaire leurs clients à moindre coût. Le prix de stockage des composants et le coût de rupture des produits finis prennent une part importante dans le prix de revient de produit et la satisfaction des clients.

Le chapitre 1 de la présente thèse s'est intéressé à présenter des généralités sur la gestion des stocks et la planification des approvisionnements des chaînes logistiques dans le cas déterministe. Nous avons également présenté les différentes incertitudes qui peuvent affecter ces méthodes de gestion et les études qui ont été intéressées à étendre ces méthodes pour le cas non déterministe.

L'état de l'art effectué au chapitre 2 a montré un manque de recherche concernant la gestion des stocks dans les chaînes logistiques lorsque les délais d'approvisionnements sont aléatoires. Ce chapitre a montré que les études qui s'intéressent à ce type d'aléa sont limitées à des structures simples de chaîne logistique à un seul niveau et une absence quasiment totale de la recherche concernant la gestion des stocks et planification d'approvisionnements lorsque la structure de la chaîne d'approvisionnements est à plus qu'un niveau, surtout pour les systèmes d'assemblage.

Les chapitres 3 et 4 avaient comme objectif de pallier à ce manque de recherche lorsque les délais d'approvisionnements suivent une loi discrète quelconque et connue. Les lois discrètes ont été choisies afin de se rapprocher de méthodes industrielles de planification du type MRP qui se situe dans un environnement temporel discret.

Ainsi, dans le chapitre 3, nous avons formulé les problèmes d'optimisation de la planification d'approvisionnements pour les systèmes à structure linéaire à multi-niveau. Nous avons étudié deux critères d'optimisation : (i) la minimisation du coût total moyen composé du coût de stockage des composants et du coût de rupture en produit fini et (ii) la minimisation du coût moyen de stockage sous la contrainte de niveau de service. Nous avons étudié deux types de situations : le cas de fabrication au plus tôt, et le cas de fabrication à une date prévue. Pour le premier cas, nous avons distingué deux problèmes P1 et P2 en étudiant les deux critères ci-dessus. Pour le deuxième cas, nous avons également considéré les deux critères et donc les deux problèmes : P3 et P4. Pour ces quatre problèmes, nous avons présenté les modèles mathématiques et une solution optimale pour chacun d'eux ainsi que des exemples numériques afin de mieux illustrer les méthodes d'optimisation

développées. Nous avons aussi présenté des testes comparatifs avec des solutions généralement utilisées par les industriels.

Dans le chapitre 4, nous nous sommes intéressés aux problèmes de planification de systèmes d'assemblage à deux niveaux. Nous avons étudié deux critères pareils à ceux du chapitre 3, nous distinguons deux problèmes en étudiant ces deux critères : PP1 et PP2. Nous avons modélisé les critères avec des expressions explicites en fonctions des dates de lancement des ordres. Nous avons ainsi obtenu deux fonctions objectifs non linéaires à variables entières. Nous avons développé un pré-traitement pour la réduction de l'espace de recherche. Pour chacun de deux problèmes, nous avons démontré des propriétés de dominances, des bornes inférieures et une borne supérieure. Ceci a permis d'utiliser des procédures par séparation et évaluation (PSE). Nous avons aussi étudié la performance des algorithmes PSE, les tests ont montré la capacité de ces algorithmes à traiter les problèmes de petite taille.

Dans le chapitre 5, nous avons présenté une méta-heuristique basée sur un algorithme génétique qui permet de résoudre les problèmes de planification de systèmes d'assemblage à deux niveaux de taille importante (le problème PP1 du chapitre 4) dans un temps très réduits toute en garantissant la qualité de la solution. Nous avons intégré une méthode de recherche locale dans l'algorithme génétique proposé. Les résultats numériques ont montré l'efficacité de notre algorithme génétique ainsi que la capacité de la recherche locale d'accélérer sa convergence.

Les résultats obtenus tout au long de ces 5 chapitres nous permettent de dégager plusieurs perspectives. D'abord, nous pouvons étendre les modèles proposés au cas de demande aléatoire. Ensuite, dans le chapitre 5, nous avons proposé un algorithme génétique. Les algorithmes génétiques sont très bien adaptés au traitement d'un problème d'optimisation multiobjectif. Le choix un algorithme génétique multiobjectif pour résoudre le problème PP2 bi-objectif (minimisation de coût de stockage et maximisation de niveau de service) peut être une autre perspective intéressante. Cette idée, qui nous paraît intéressante, a un avantage qu'elle permettrait aux industriels d'avoir un ensemble de compromis entre le coût de stockage et le niveau de service et donc une flexibilité de décision.

A court terme notre défi sera d'exploiter ces perspectives et de développer les algorithmes d'optimisation pour chacun des problèmes présentés au cas de demande aléatoire. A moyen terme, nous intéresserons à intégrer de la simulation dans les méthodes d'optimisation proposées afin de pouvoir étendre notre étude aux cas des systèmes d'assemblages multi-niveaux (avec plus de deux niveaux).

# Bibliographie

- [1] Arda Y., Henet J.C., (2006). “Inventory control in a multi-supplier system”. *International Journal of Production Economics*, 104 (2), pp. 249-259.
- [2] Axsäter S., (2005). “Planning order releases for an assembly system with random operation times”. *Operations Research Spectrum*, 27, pp. 459-470.
- [3] Axsäter S., (2006). *Inventory control*, 2<sup>nd</sup> ed., Springer.
- [4] Bagchi U., Hayya J.C., Chu C.H., (1986). “The effect of lead time variability. The case of independent demand”. *Journal of Operations Management*, 6 (2), pp. 159-177.
- [5] Bäck T., (1994). “Selective pressure in evolutionary algorithms: A characterization of selection mechanisms”. In *Proc. 1st IEEE Conf. on Evolutionary Computation*. Piscataway, NJ: IEEE Press, pp. 57–62.
- [6] Bookbinder J.H., Çakanyildirim M., (1999). “Random lead-times and expedited orders in  $(Q,r)$  inventory systems”. *European Journal of Operational Research*, 115, pp. 300-313.
- [7] Borisovsky P., Dolgui A., Ereemeev A., (2007). “Genetic algorithms for a supply management problem: MIP-recombination vs greedy decoder”. *European Journal of Operational Research*, In Press, Corrected Proof, Available online 19 November 2007.
- [8] Bradley J. R., Robinson L. W., (2005). “Improved base-stock approximations for independent stochastic lead times with order crossover”. *Manufacturing & Service Operations Management*, 7(4), pp. 319-329.
- [9] Çakanyildirim M., Bookbinder J., Gerchak Y., (2000). “Continuous review inventory models where random lead time depends on lot size and reserved capacity”. *International Journal of Production Economics*, 68 (3), pp. 217-228.
- [10] Chen C., Vempati V.S., Aljaber N., (1995). “An application of genetic algorithms for flow shop problems”. *European Journal of Operational Research*, 80, pp. 389-396.
- [11] Chen R.-S., Lu K.-Y., Yu S.-C., (2002). “A hybrid genetic algorithm approach on multi-objective of assembly planning problem”. *Engineering Applications of Artificial Intelligence*, 15 (5), pp. 447-457.
- [12] Chiang C., Benton W.C., (1994). “Sole sourcing versus dual sourcing under stochastic demands and lead times”. *Naval Research Logistics*, 41, pp. 609-624.

- [13] Chu C., Proth J.M., Xie X., (1993). "Supply management in assembly systems". *Naval Research Logistics*, 40, pp. 933-949.
- [14] De Bodt M.A., Van Wassenhove L.N., (1983). "Cost Increases Due to Demand Uncertainty in MRP Lot Sizing". *Decision Science*, 14, pp. 345-362.
- [15] DeMatteis J.J., (1968). "An Economic Lot-Sizing Technique I: The Part-Period Algorithm". *IBM Systems Journal*, 7 (1), pp. 30-38.
- [16] Ding H., Benyoucef L., Xie X., (2006). "A simulation-based multi-objective genetic algorithm approach for networked enterprises optimization". *Engineering Applications of Artificial Intelligence*, 19, pp. 609-623.
- [17] Dirickx M.I., Koevoets D., (1977). "A continuous reviews inventory model with compound poisson demand process and stochastic lead time". *Naval Research Logistics Quarterly*, 24(4), pp. 577-585.
- [18] Dolgui A., Portmann M.C., Proth J.M., (1995). "Planification de systèmes d'assemblage avec approvisionnement aléatoire en composants". *Journal of Decision Systems*, 4 (4), pp. 255-279.
- [19] Dolgui A., and Ould Louly M.A., (2002). "A Model for Supply Planning under Lead Time Uncertainty". *International Journal of Production Economics*, 78, pp. 145-152.
- [20] Dolgui A., Ereemev A., Kolokolov A., Sigaev V., (2002). "Buffer allocation in production line with unreliable machines". *Journal of Mathematical Modeling and Algorithms*, 2, pp. 89-104.
- [21] Dolgui A., Proth J.M. (2006). *Les systèmes de production modernes*. Tome 1 : Conception, gestion et optimisation. Hermès, 2006, (ISBN: 2-7462-1249-8), 415 pages.
- [22] Dolgui A., Proth J.M. (2006). *Les systèmes de production modernes*. Tome 2 : Outils et corrigés des exercices. Hermès, 2006, (ISBN: 2-7462-1250-1), 385 pages.
- [23] Dolgui A., Prodhon C., (2007). "Supply planning under uncertainties in MRP environments: a state of the art". *Annual Reviews in Control*, 31, pp. 269-279.
- [24] Dolgui A., Ereemev A.V., Sigaev V.S., (2007). "HBBA: Hybrid Algorithm for Buffer Allocation in Tandem Production Lines". *Journal of Intelligent Manufacturing*, 18 (3), pp. 411-420.
- [25] Dupont L., (1998). *La gestion industrielle*. Ed. Hermès, Paris, 1998.
- [26] Ehrhardt R. (1984). " $(s, S)$  policies for a dynamic inventory model with stochastic lead times". *Operations Research*, 32, pp. 121-132.

- [27] Elhafsi M., (2002). "Optimal leadtimes planning in serial production systems with earliness and tardiness costs". *IIE Transactions*, 34, pp. 233-243.
- [28] Enns S.T., (2002). "MRP performance effects due to forecast bias and demand uncertainty". *European Journal of Operational Research*, 138(1), pp. 87-102.
- [29] Eppen G.D., Martin R.K., (1988). "Determining safety stock in the presence of stochastic lead time and demand". *Management Science*, 34, pp. 1380-1390.
- [30] Fong D.K.H., (1992). "A note on exact moment computation for normal lead times in the two-supplier case". *Journal of Operational Research Society*, 43 (1), pp. 63–69.
- [31] Fong D.K.H., Gempesaw, V.M., Ord J.K., (2000). "Analysis of a dual sourcing inventory model with normal unit demand and Erlang mixture lead times". *European Journal of Operational Research*, 120, pp. 97-107.
- [32] Fujiwara O., Sedarage D., (1997). "An optimal (Q,r) policy for a multipart assembly system under stochastic part procurement lead times". *European Journal of Operational Research*, 100, pp. 550-556.
- [33] Georghios P., Sphicas G.P.,(1982). "On the Solution of an Inventory Model with Variable Lead Times". *Operations Research*, 30(2), pp. 404-410.
- [34] Giard V., (2003). *Gestion de la production et des flux*, 3<sup>ème</sup> Edition. Economica, Paris, 2003.
- [35] Golberg D.E., (1989). "Genetic Algorithms in Search, Optimisation and Machine Learning". Addison Wesley, Reading, MA.
- [36] Goldberg D.E., Deb K., (1991). "A comparative analysis of selection schemes used in genetic algorithms". In : *Foundations of Genetic Algorithms*. San Mateo, CA: Morgan Kaufmann, pp. 69–93.
- [37] Grasso, E.T., Taylor B.W., (1984). "A Simulation-Based Experimental Investigation of Supply/Timing Uncertainty in MRP Systems". *International Journal of Production Research*, 22 (3), pp. 485-497.
- [38] Graves S.C, (1988). "Safety stocks in manufacturing systems". *Journal of Manufacturing and Operations Management*, 1, pp. 67-101.
- [39] Gomaa A.H., Hussien S.M., Zahran M.Z., "A simulation study for MRP system under uncertainty operating environment". *Journal of Engineering and Applied Science*, 1999, 46(2), pp. 237-256.
- [40] Gross D., Harris C.M., Lechner, J.A., (1971). "Stochastic Inventory Models with Bulk Demand and State-Depend Leadtimes". *Journal of Applied Probability*, 8, pp. 521-534.

- [41] Gross D., Harris C.M., (1973). “Continuous-Reviews (s, S) Inventory Models with State-Depend Leadtimes”. *Management Science*, 19, pp. 567-574.
- [42] Guo Y., Ganeshan R., (1995). “Are more suppliers better?”. *Journal of Operational Research Society*, 46 (5), pp. 892–896.
- [43] Gupta S.M., Brennan L., (1995). « MRP systems under supply and process uncertainty in an integrated shop floor control”. *International Journal of Production Research*, 33, pp. 205-220.
- [44] Gurnani H., Akella R., Lehoczky J., (1996). “Optimal order policies in assembly systems with random demand and random supplier delivery”. *IIE Transactions*, 28, pp. 865-878.
- [45] Hadley G., Whitin T.M., (1963). *Analysis of Inventory Systems*, Prentice-Hall, Englewood Cliffs, NJ.
- [46] Hansmann F., (1959). “Optimal inventory location and control in production and distribution networks”. *Operations Research*, 7, pp. 483-498.
- [47] Harris, F. W., (1913). “How many parts to make at once”. *The Magazine of Management*, 10, pp. 135-136.
- [48] Harrison J.M., (1973). “Assembly-Like Queues”. *Journal Applied Probability*, 10, pp. 354-367.
- [49] Hayya J.C., Xu S.H., Ramasesh R.V., He X.X., (1995). “Order crossover in inventory systems”. *Stochastic Models*, 11 (2), pp. 279–309.
- [50] He X.J., Kim J.G., Hayya J. C. (2005). “The cost of lead-time variability: The case of the exponential distribution”. *International Journal of Production Economics*, 97(2), pp. 130-142.
- [51] Hegedus M.G., Hopp W.J., (2001). “Setting procurement safety lead-times for assembly systems”. *International Journal of Production Research*, 39(15), pp. 3459-3478.
- [52] Ho C.J., Lau H.S., (1994). “Evaluating the Impact of Lead Time Uncertainty in Material Requirements Planning Systems”. *European Journal of Operational Research*, 75, pp. 89-99.
- [53] Ho C.J., Ireland T.C., (1998). “MRP system nervousness and forecast errors”. *International Journal of Production Research*, 36, pp. 2285-2299.
- [54] Holland J.H., (1975). “Adaptation in Natural and Artificial Systems”. *University of Michigan Press: Ann Arbor, MI*.
- [55] Houpt R.L. , Houpt S.E., (1998). “Practical genetic algorithms”, Wiley, New York, NY.

- [56] Inderfurth, K., (2007). "How to protect against demand and yield risks in MRP systems". *International Journal of Production Economics*, doi :10.1016/j.ijpe.2007.02.005.
- [57] Jeunet J., Jonard N., (2000). "Measuring the Performance of Lot-Sizing Techniques in Uncertain Environments". *International Journal of Production Economics*, 64, pp. 197-208.
- [58] Kalpakam S., Shahul Hameed M.A., (1981). "Optimum ordering policies with random lead times". *Micrelectron Reliab*, 21(5), pp. 737-741.
- [59] Kaplan R.S., (1970). "A Dynamic Inventory Model with Stochastic Lead Times". *Management Science*, 16, pp. 491-507.
- [60] Kim J.G., Sun D., He X.J., Hayya J.C., (2004). "The (s,Q) inventory model with Erlang lead time and deterministic demand". *Naval Research Logistics*, 51, pp. 906-923.
- [61] Kim J.G, Chatfield D., Harrison T., Hayya J., (2006). "Quantifying the bullwhip effect in a supply chain with stochastic lead time". *European Journal of Operational Research*, 173, pp. 617-636.
- [62] Kumar A., (1989). "Component Inventory Cost in an Assembly Problem with Uncertain Supplier Lead-Times". *IIE Transactions*, 21 (2), pp. 112-121.
- [63] Lee H.L., Billington C., (1993). "Material management in decentralized supply chains". *Journal of Operations Research*, 41 (5), pp. 835-847.
- [64] Lee H.L., Nahmias S., (1993). "Single-Product, Single-Location Models". In: *Logistics of Production and Inventory* (S.G. Graves, A.H.G. Rinnooy Kan, P.H. Zipkin, Eds.), Handbooks in Operations Research and Management Science, vol. 4, pp. 3-55. Elsevier Science, Amsterdam.
- [65] Liberatore M.J., (1979). "The EOQ Model under Stochastic Lead Times". *Operations Research*, 27, pp. 391-396.
- [66] Maloni M.J., Benton W.C., (1997). "Supply chain partnerships: opportunities for operations research". *European Journal of Operational Research*, 101, pp. 419-429.
- [67] Melnyk S.A., Piper C.J., (1981). "Implementation of material requirements planning". *International Journal of Production and Operations Management*, 2, pp. 52-60.
- [68] Michalewicz Z., Fogel D.B., (2002). *How to Solve It: Modern Heuristics*, third ed. Springer, Berlin.
- [69] Miller J.G., (1979). "Hedging the master shedule". In: Ritzman L.P., Krajewski L.J., Berry W.L., Goodman S.H., Hardy S.T., Vitt L.D.(eds.), *Disaggregation problems in manufacturing and service organizations*. Martinus Nijhoff, Boston, MA, pp. 237-256.

- [70] Minner S., (2003). "Multiple-supplier inventory models in supply chain management: a review". *International Journal of Production Economics*, 81-82, pp. 265-279.
- [71] Mohebbi E., Posner M.J.M., (1998). "Sole versus dual sourcing in a continuous-review inventory system with lost sales". *Computers in Industrial Engineering*, 34, pp. 321-336.
- [72] Mohebbi E., (2003). "Supply interruptions in a lost-sales inventory system with random lead time". *Computers & Operations Research*, 30, pp. 411-426.
- [73] Mollet H., Ballot E., Dutreuil J., Fontaine F., (2006). *Système de production et de logistique*. Ed. Hermès, Paris, 2006.
- [74] Molinder A., (1997). "Joint optimisation of lot-sizes, safety stocks and safety lead times in an MRP system". *International Journal of Production Research*, 35, pp. 983-994.
- [75] Moscato P., (2000). "Memetic algorithms". In P. Pardalos and M. Resende, *Handbook of Applied Optimization*, Oxford University Press, Oxford.
- [76] Mula J., Poler R., Garcia-Sabater J.P., Lario F.C., (2006). "Models for production planning under uncertainty: A review". *International Journal Production Economics*, 103, pp. 271-285.
- [77] Nahmias S., (1979). Simple approximations for a variety of dynamic leadtime lost-sales inventory models. *Operation Research*, 46, pp. 5112-5119.
- [78] Nahmias, S., (1997). *Production and Operations Analysis*. Irwin.
- [79] Namit K., (2007). "Determining reorder point in the presence of stochastic lead time and Box-Jenkins time series demand". *Journal of the Academy of Business and Economic*, 7(2), pp. 208-215.
- [80] Ould Louly M.-A., (2001). "Optimisation des stocks des composants et planification de leurs réapprovisionnements pour les systèmes d'assemblage soumis à des aléas". Thèse de doctorat de l'UTT (Université de Technologie Troyes), directeurs de thèse : Alexandre Dolgui.
- [81] Ould Louly M.-A., Dolgui A., (2002). "Generalized newsboy model to compute the optimal planned lead times in assembly systems". *International Journal of Production Research*, 40 (17), pp. 4401-4414.
- [82] Ould-Louly M.-A., Dolgui A., (2004). "The MPS parameterization under lead time uncertainty". *International Journal of Production Economics*, 90(3), 18, pp. 369-376.
- [83] Ould Louly M.-A., Dolgui A., (2008). "Calculating Safety Stocks for Assembly Systems with Random Component Procurement Lead Times: Branch and Bound Algorithm". *European Journal of Operational Research*, Available online (Science Direct).

- [84] Ould Louly M.-A., Dolgui A., Hnaien F., (2008 a). "Optimal Supply Planning in MRP Environments for Assembly Systems with Random Component Procurement Times". *International Journal of Production Research*, 46(19), pp. 5441-5467.
- [85] Ould Louly M.-A., Dolgui A., Hnaien F., (2008 b). "Supply planning for single-level assembly system with stochastic component delivery times and service level constraint". *International Journal of Production Economics*. Available online 5 juillet 2008 (Science Direct).
- [86] Pan A.C., Ramasesh R.V., Hayya J.C., Ord J.K., (1991). "Multiple sourcing: the determination of lead times". *Operations Research Letters*, 10, pp. 1-7.
- [87] Pasandideh S.H.R., Niaki S.T.A., (2008). "A genetic algorithm approach to optimize a multi-products EPQ model with discrete delivery orders and constrained space". *Applied Mathematics and Computation*, 195, pp. 506-514.
- [88] Plenert G., (1999). "Focusing Material Requirements Planning (MRP) Towards Performance". *European Journal of Operational Research*, 119, pp. 91-99.
- [89] Porteus E.L., (1990). "Stochastic inventory theory". In: Hand-book in OR and MS, Vol 2, Heyman D.P. and Sobel M.J. (eds.). Elsevier Science Publishers, pp. 605-652.
- [90] Prins C., (2000). "Competitive genetic algorithms for the open-shop scheduling problem". *Mathematical Methods of Operations Research*, 52, pp. 389-411.
- [91] Proth J.M., Mauroy G., Wardi Y., Chu C., Xie X., (1997). "Supply management for cost minimization in assembly systems with random components yield times". *Journal of Intelligent Manufacturing*, 8, pp. 385-403.
- [92] Ramasesh R.V., Ord J.K., Hayya J.C., Pan A., (1991). "Sole versus dual sourcing in stochastic lead-time ( $s, Q$ ) inventory models". *Management Science*, 37 (4), pp. 428-443.
- [93] Ramasesh R.V., Ord J.K., Hayya J.C., (1993). "Note: dual sourcing with non-identical suppliers". *Naval Research Logistics*, 40, pp. 279-288.
- [94] Riezebos J., (2006). "Inventory order crossovers". *International Journal of Production Economics*, 104, pp. 666-675.
- [95] Robinson L.W., Bradley J.R., Thomas L.J., (2001). "Consequences of order crossover under order-up-to inventory policies". *Manufacturing & Service Operations Management*, 3 (3), pp. 175-188.
- [96] Sculli D., Wu S.Y (1981). "Stock control with two suppliers and normal lead times", *Journal of the Operational Research Society*, 32, pp. 1003-1011.

- [97] Sculli D., Shum Y.W., (1990). "Analysis of a continuous review stock-control model with multiple suppliers". *Journal of the Operational Research Society*, 41(9), pp. 873-877.
- [98] Sedarage D., Fujiwara, O., H.T. Luong, (1999). "Determining optimal order splitting and reorder level for  $N$ -supplier inventory systems". *European Journal of Operational Research*, 116, pp. 389-404.
- [99] Silver E.A., Meal H.C., (1973). "A heuristic for selecting lot-sizing quantities for the case of deterministic time-varying demand rates and discrete opportunities for replenishment". *Production and Inventory Management*, 14, pp. 64-74.
- [100] Silver E.A., Zufferey N., (2005). "Inventory control of raw materials under stochastic and seasonal lead times". *International Journal of Production Research*, 43, pp. 5161-5179.
- [101] Simpson K.F, (1958). "In-process inventories". *Operations Research*, 6, pp. 863-873.
- [102] Som P., Wilhelm W.E., Disney R.L., (1994). "Kitting process in a stochastic assembly system". *Queueing Systems*, 17, pp. 471-490.
- [103] Song J.S., Yano C.A., Lerssrisuriya P., (2000). "Contract assembly: Dealing with combined supply lead time and demand quantity uncertainty". *Manufacturing and Service Operations Management*, 2(3), pp. 287-296.
- [104] Sphicas G.P., 1982. "On the solution of an inventory model with variable lead times," *Operations Research*, 30(2), pp. 404-410.
- [105] Sudhir Ryan Daniel J., Rajendran C., (2005). "A simulation-based genetic algorithm for inventory optimization in a serial supply chain". *International Transactions in Operational Research*, 12, pp. 101-127.
- [106] Tang O., Grubbström, R.W., (2003). "The detailed coordination problem in a two-level assembly system with stochastic lead times". *International Journal Production Economics*, 81-82, pp. 415-429.
- [107] Tang O., Grubbström R.W., (2005). "Considering stochastic lead times in a manufacturing/remanufacturing system with deterministic demands and returns". *International Journal of Production Economics*, 93-94, pp. 285-300.
- [108] Tang O., Grubbström R.W., Zanoni, S., (2007). "Planned lead time determination in a make-to-order remanufacturing system". *International Journal of Production Economics*, 108 (1-2), pp. 426-435.
- [109] Tijms H.C., Groenevelt H., (1984). "Simple approximations for the reorder point in periodic and continuous review (s,S) inventory systems with service level constraints". *European Journal of Operational Research*, 17, pp. 175-190.

- [110] Vollmann T.E., Berry W.L., Whybark D.C., (1997). *Manufacturing Planning and Control Systems*. Irwin/Mcgraw-Hill.
- [111] Wagner H.M., Whitin T.M., (1958). "Dynamic version of the economic lot size model". *Management Science*, 5, pp. 89-96.
- [112] Wang L., Zhang L., Zheng D.-Z., (2006). "An effective hybrid genetic algorithm for flow shop scheduling with limited buffers". *Computers & Operations Research*, 33, pp. 2960-2971.
- [113] Weeks J.K. (1981). "Optimizing Planned Lead Times and Delivery Dates". *21<sup>st</sup> annual Conference Proceeding, American Production and Inventory Control Society*, pp. 177-188.
- [114] Whybark D. C., Williams J.G., (1976). "Material Requirements Planning Under Uncertainty". *Decision Science*, 7, pp. 595-606.
- [115] Wilhelm W.E., Som P., (1998). "Analysis of a single-stage, single-product, stochastic, MRP-controlled assembly system". *European Journal of Operational Research*, 108, pp. 74-93.
- [116] Yano C.A., (1987 a). "Setting planned leadtimes in serial production systems with tardiness costs". *Management Science*, 33(1), pp. 95-106.
- [117] Yano C.A., (1987 b). "Planned leadtimes for serial production systems". *IIE Transactions*, 19(3), pp. 300-307.
- [118] Yano C.A., (1987c). "Stochastic leadtimes in two-level assembly systems". *IIE Transactions*, 19(4), pp. 95-106.
- [119] Zalkind, D. (1978). "Order –level inventory systems with independent stochastic leadtimes". *Management Science*, 24(13), pp. 1384-1392.
- [120] Zangwill W.I. (1966). "A deterministic multi-period production scheduling model with backlogging". *Management Science*, 13, pp. 105-119.
- [121] Zheng Y.S., (1992). "On properties of stochastic inventory systems". *Management Science*, 28 (1), pp. 87-103.
- [122] Zipkin P., (1986). "Stochastic leadtimes in continuous-time inventory models". *Naval Research Logistics Quarterly*, 33, pp. 763-774.
- [123] Zipkin P., (2000). *Foundation of Inventory Management*, McGraw-Hill, USA.

## **Résumé**

La gestion des stocks est un élément très important pour les entreprises. Il faut pouvoir satisfaire les clients à moindre coût. Pour cela, il est nécessaire d'être en possession de tous les composants, pour fabriquer les produits demandés et les livrer à la date voulue. En effet, une mauvaise politique d'approvisionnement en composants conduit soit à des retards de livraison, qui engendrent des frais, soit à des stocks inutiles. Dans cette thèse, nous étudions la planification des réapprovisionnements en composants pour les systèmes de production de différentes structures. Nous travaillons avec les nomenclatures à plusieurs niveaux et nous tenons compte des aléas des délais d'approvisionnement. Nous avons choisi comme variables de décision celles qui correspondaient aux paramètres de la méthode MRP, comme le temps de cycle planifié. L'objectif pratique de notre étude étant également de fournir des techniques pour le paramétrage des logiciels MRP en présence de ce type d'aléas. Nous avons proposé des modèles et avons démontré leurs propriétés théoriques intéressantes ainsi que des approches d'optimisation nouvelles pour ce type de problème.

**Mots clés :** gestion des stocks, planification des approvisionnements, délais d'approvisionnement aléatoires, PSE, algorithme génétique.

## **Abstract:**

The inventory control is crucial for companies who wish to satisfy their customer demands on time as well as controlling costs. For this it is necessary to be in possession of the necessary and sufficient components to produce the requested products on time. Indeed, a poor inventory control policy leads to overstocking or stockout situations. In the former, the generated inventories are expensive and in the latter there are shortages and penalties due to unsatisfied customer demands. In this thesis, we study the supply planning for the multilevel supply chains. The decision variables are the parameters of MRP (planned lead times). The aim is to provide ideas and methods on the MRP parameterization under lead time uncertainties. We proved interesting theoretical properties and we proposed the models and methods for the optimization problems.

**Keywords:** inventory control, supply planning, random lead times, B&B, genetic algorithm.

N° d'ordre : 502 GI

**Prénom NOM : Faicel Hnaien**

**DISSERTATION TITLE :**

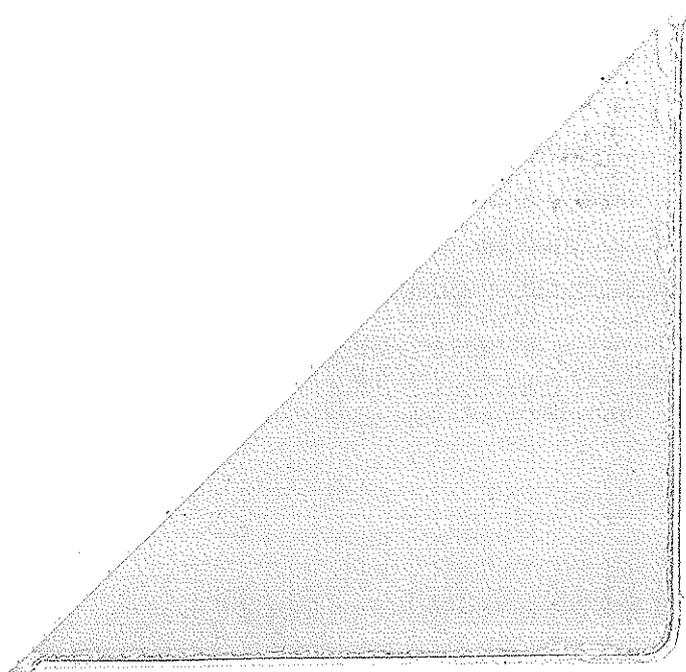
Inventory control for supply chains under lead time uncertainties

Speciality : Industrial engineering

Keywords : inventory control, supply planning, random lead times, B&B, genetic algorithm.

**Abstract :**

The inventory control is crucial for companies who wish to satisfy their customer demands on time as well as controlling costs. For this, it is necessary to be in possession of the necessary and sufficient components to produce the requested products in time. Indeed, a poor inventory control policy leads to overstocking or stockout situations. In the former, the generated inventories are expensive and in the latter, there are shortages and penalties due to unsatisfied customer demands. In this thesis, we study the supply planning for the multilevel supply chains. The decision variables are the parameters of MRP (planned lead times). The aim is to provide ideas and methods on the MRP parameterization under lead time uncertainties. We proved interesting theoretical properties and we proposed the models methods for the optimization problems.



N° d'ordre : 502 GI

**Prénom Nom : Faicel Hnaien**

**TITRE DE LA THÈSE :**

Gestion des stocks Dans Des chaînes logistiques face aux aléas des délais d'approvisionnements

Spécialité : Génie Industriel

Mots clefs : gestion des stocks, planification des approvisionnements, délais d'approvisionnement aléatoires, PSE, algorithme génétique.

**Résumé :**

La gestion des stocks est un élément très important pour les entreprises. Il faut pouvoir satisfaire les clients à moindre coût. Pour cela, il est nécessaire d'être en possession de tous les composants, pour fabriquer les produits demandés et les livrer à la date voulue. En effet, une mauvaise politique d'approvisionnement en composants conduit soit à des retards de livraison, qui engendrent des frais, soit à des stocks inutiles. Dans cette thèse, nous étudions la planification des réapprovisionnements en composants pour les systèmes de production de différentes structures. Nous travaillons avec les nomenclatures à plusieurs niveaux et nous tenons compte des aléas des délais d'approvisionnement. Nous avons choisi comme variables de décision celles qui correspondaient aux paramètres de la méthode MRP, comme le temps de cycle planifié. L'objectif pratique de notre étude étant également de fournir des techniques pour le paramétrage des logiciels MRP en présence de ce type d'aléas. Nous avons proposé des modèles et avons démontré leurs propriétés théoriques intéressantes ainsi que des approches d'optimisation nouvelles pour ce type de problème.

---